

# **SKRIPSI**

## **SISTEM ABSENSI PADA MURID SEKOLAH MENGGUNAKAN RFID DENGAN TAMPILAN LCD BERBASIS MIKROKONTROLER AT89S8252 YANG DIINFORMASIKAN KE WALI MURID VIA SMS**



**Disusun Oleh :  
LISTYAMI FATMAWATI  
NIM : 03.17.110**



**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**SEPTEMBER 2007**

---

## LEMBAR PERSETUJUAN

**SISTEM ABSENSI PADA MURID SEKOLAH MENGGUNAKAN RFID  
DENGAN TAMPILAN LCD BERBASIS MIKROKONTROLER  
AT89S8252 YANG DIINFORMASIKAN KE WALI MURID VIA SMS**

### SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

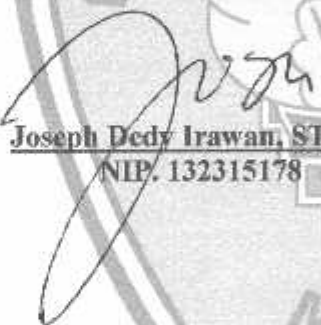
**Disusun Oleh :**

**LISTYAMI FATMAWATI  
NIM : 03.17.110**

**Diperiksa dan Disetujui**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

  
**Joseph Dedy Irawan, ST, MT  
NIP. 132315178**

  
**Ir. Eko Nurcahyo  
NIP.Y 1028700172**

**Mengetahui**

**Ketua Jurusan Teknik Elektro S-1**

  
**Ir. F. Yudi Limpraptono, MT  
NIP.Y 1039500274**

**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2007**



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA

---

**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Listyami Fatmawati  
NIM : 03.17.110  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Sistem Absensi Pada Murid Sekolah Menggunakan RFID  
Dengan Tampilan LCD Berbasis Mikrokontroler AT89S8252  
Yang Diinformasikan Ke Wali Murid Via SMS.

Dipertahankan di hadapan majelis penguji Skripsi jenjang Strata satu ( S-1 ) pada :

Hari : Senin  
Tanggal : 4 September 2007  
Dengan Nilai : 88,6 (A) *84*



**Ketua Majelis Penguji**

**(Ir. Mochtar Asroni, MSME)**  
NIP. Y. 1018100036

**Sekretaris Majelis Penguji**



**(Ir. F. Yudi Limpraptono, MT)**  
NIP. Y. 1039500274

**Penguji I**



**(Ir. Widodo Pudji Mulyanto, MT)**  
NIP. Y. 1028700171

**Penguji II**



**(I Komang Somawirata, ST, MT)**  
NIP. P. 1030100361

---

## ABSTRAKSI

### **SISTEM ABSENSI PADA MURID SEKOLAH MENGGUNAKAN RFID DENGAN TAMPILAN LCD BERBASIS MIKROKONTROLER AT89S8252 YANG DIINFORMASIKAN KE WALI MURID VIA SMS**

**Listyami Fatmawati**

**03.17.110**

**Jurusan Teknik Elektronika – Institut Teknologi Nasional Malang**

**Jln. Raya Karanglo Km 2 Malang**

**simplegirl11tya@gmail.com**

**Dosen Pembimbing : I. Joseph Dedy Irawan, ST, MT.**

**II. Ir. Eko Nurcahyo**

**Kata Kunci : RFID, Mikrokontroler, SMS.**

Penerapan sikap disiplin dalam lingkungan sekolah memegang peranan penting dalam upaya pencapaian tujuan pembelajaran disekolah. Salah satunya adalah kedisiplinan kehadiran murid disekolah.

Dari fenomena yang ada, permasalahan yang sering terjadi dimana murid yang tidak masuk sekolah tanpa sepengetahuan orang tua serta tidak ada ijin dari pihak lembaga sekolah dengan artian murid tersebut membolos sekolah. Untuk itu pihak lembaga sekolah perlu adanya suatu sistem yang dapat mempermudah dalam proses absensi murid di sekolah serta sistem informasi yang bisa disampaikan ke orang tua wali murid secara otomatis.

Dengan memanfaatkan Teknologi Tag RFID (*Radio Frequency Identification*) sebagai kartu absensi, kemudian dengan bantuan RFID reader beserta mikrokontroler AT89S8252, dan LCD, tag tersebut dapat diidentifikasi. Untuk menampilkan data dari murid tersebut, berdasarkan kode dari Tag RFID yang telah teridentifikasi dilakukan dengan bantuan komputer. Kemudian setelah proses absensi dengan menggunakan teknologi *celluler* yang sudah populer didunia yaitu layanan *messaging* yang tak lain adalah SMS (*Short Message Service*) sebagai media informasi kehadiran murid di sekolah kepada masing-masing wali murid.



## KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran-Mu Ya Allah yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat menyelesaikan skripsi yang berjudul “Sistem Absensi Pada Murid Sekolah Menggunakan RFID Dengan Tampilan LCD Berbasis Mikrokontroler AT89S8252 Yang Diinformasikan Ke Wali Murid Via SMS” ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
2. Bapak Joseph Dedy Irawan, ST, MT selaku Dosen Pembimbing I dan Ka. Laboratorium Elektronika Digital.
3. Bapak Ir. Eko Nurcahyo selaku Dosen Pembimbing II dan Ka. Laboratorium Perancangan Elektronika.
4. Ayah dan Ibu serta saudara-saudara kami yang telah memberikan do’a restu, dorongan, semangat, dan biaya.
5. Rekan-rekan Instruktur di Laboratorium Perancangan Elektronika.
6. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penyusun telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu

penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, September 2007

***Penyusun***

# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah...wasyukurillah....ku bersyukur pada-Mu ya Allah atas rahmat dan hidayah yang Engkau berikan, sehingga aku bisa menyelesaikan skripsi ini dengan baik dan lancar. Serta salam dan solawat atas junjungan Nabi Muhammad Saw.

Sungkem kagem Nyak and Babe Tercinta,,,, Alhamdulillah Tya udah Sarjana Teknik... Terima Kasih atas Segala doa restu, semangat, dan tak ketinggalan juga Materi tentunya, yang selama ini tiada henti-hentinya kalian berikan. Tya tahu masih belum bisa membalas segala pengorbanan Nyak & Babe yang mungkin tiada tergantikan oleh apapun. Sekali lagi....Tya ucapken....*Xiexie*...

Salam hangat dan sayang for my family.....Buat my sister i2k & mas Iparku ( kadek...) yang selalu kasih *support* & nasehat2 yang selalu bikin aku jadi lebih percaya diri...suwun yo...,and buat mbak idha....yang selalu menemani aku tidur...( hobinya tidur lo...), nyuciin bajuku...nemenin aku belanja...thengkyu yo...mbrur...eh o..iya buat Kucingarongmu ( suAndik...) makasih ya sudah jadi teman berantem yang buaek & met Lulus jg yo. Kagem Mbah Yi...Makasih doa - doanya...ampuh bgt...semuanya jadi luancarr. Trus buat bhi...terimakasih...udah nemenin ney selalu...,mg cepet selesai & lancar skripsinya ya...,buat mbak&mas nya serta buat oiy yg makin lucu...maturthengkyu.

Wahai Saudaraku Inul&Intan...yang selalu berbagi susah maupun senang....makasih ya...,Tya gak bakal lupa...semua kenangan ma klian...menyenangkan...Inget ya ITI Forever...

Buat keluarga Besar Lab, Perancangan Elektronika buat Pak Eko selaku Kepala Lab dan juga Dosen pembimbing Tya..makasih ya Pak...atas segala nasehat dan bantuannya. Serta Buat All Instruktur...Mas Ditto, Sonni, Rico, Neal ajie, Putri & Jonathan ( thank's backgroundnya ) yg rukun ya, Micko, Erric, Hamzah, Riza, Rozzi, Andri, yunink dan semua yang belum tersebutkan makasih buat dukungannya...Berikan yang terbaik buat Lab kita...dan tetap menjadi yang terdepan. Mas Tatan. yang selalu melindungi aku&inul...dari segala mara bahaya..he..sukses buat sampeyan mas. And Buat Bang Faisol..Bang..Aries...makasih buanyaldd mas ya...atas.....Ilmunya...bantuannya...leluconya...smuaannya...deh...Moga Sukses selalu Buat klian b'dua. Mas Very&Adeknya...suwun yo dukungannya..moga klian langgeng..he... buat Huda&m'deny walupun dah g krenu lg tp tya pengen ngucap muakasih mas ya...dlu sampeyan sering bantu aku...terutama pas pkn dlu... Tyuzz mas cs (m'middy et all) yg slalu bikin kinclong lantai..bs pakek ngaca donk..he..makasih ya buat obrolannya klo tya blum dapat kunci lab...matursuwun.

Dosen Pembimbing aku Pak Joseph...makasih atas bimbinganya dan bantuannya... Buat mama Irrine and mama Irma..makasih ya mam's selalu kasih semangat buat Tya... Buat teman - teman Elka '03 semuanya...makasih udah 4 taon ini jadi teman seperjuangan kuliah...sukses..buat kalian yo... Eko sobat yg buaek mat lulur juga ya... Hafiz&Ony...diselesein skripsinya ya. Jng manja..ya..he To teman2 seperjoeangan skripsi..M'Indra..Mb'pi2n M'Udin..M'Andik..Hendrick..Candra..Akhirnya..ST..juga Makasih buat semangat kebersamaan, serta guyonan..nya... Ok guys...moga sukses buat klian semua..Amin.

Arek2 290C m'yantie. ita san yanie..thank's.. buat rumpian&candaannya...p'ju..makasih udah sering.. bukaln pintu klo tYa pulang malem...hicks and thank's to crew ngopie-Net and Boz Anto' makasih dah ngasih kesempatan buat gabung...and nambah ilmu dech... yg terakhir Thank's to...DraFanta&Lasimin Family...

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>i</b>
<b>ABSTRAKSI .....</b>	<b>ii</b>
<b>KATA PENGANTAR .....</b>	<b>iii</b>
<b>DAFTAR ISI .....</b>	<b>v</b>
<b>DAFTAR GAMBAR .....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan .....	3
1.5. Metodologi Penulisan .....	3
1.6. Sistematika Penulisan.....	4
<b>BAB II TEORI DASAR .....</b>	<b>6</b>
2.1. Pendahuluan .....	6
2.2. RFID ( <i>Radio Frequency Identification</i> ) .....	6
2.2.1. <i>Komponen RFID Tag</i> .....	6
2.2.1.1. <i>Tag</i> .....	7
2.2.1.2. <i>Tag Reader</i> .....	7
2.2.1.3. <i>Server Database</i> .....	9
2.2.2. <i>Mekanisme RFID</i> .....	9

2.2.2.1. Prinsip Kerja <i>Reader</i> dan <i>Tag/Transpoder</i> .....	9
2.2.2.2. Pembacaan .....	11
2.2.2.3. Modulasi Yang Digunakan .....	12
2.2.2.4. Pengiriman Data .....	14
2.2.2.5. Format Pembacaan ASCII .....	14
2.3. Mikrokontroler ATMEL AT89S8252 .....	15
2.3.1. Fasilitas Mikrokontroler AT89S8252 .....	15
2.3.2. Pinout Mikrokontroler AT89S8252 .....	18
2.3.3. SFR tambahan pada mikrokontroler Atmel AT89S8252 .....	21
2.3.3.1. SFR untuk Timer 2 .....	22
2.3.3.2. SFR Pengontrol SPI .....	24
2.3.4. Organisasi Memori .....	29
2.3.5. Timer 2 .....	30
2.3.5.1. Mode <i>Capture</i> .....	31
2.3.5.2. <i>Auto Reload (Up/Down Counter)</i> .....	32
2.3.5.3. <i>Baud Rate Generator</i> .....	34
2.3.5.4. <i>Programmable Clock Out</i> .....	35
2.3.6. Sistem Interupsi .....	36
2.4. <i>Interface Serial</i> .....	37
2.4.1. Komunikasi Serial Antara PC dengan MCU .....	37
2.4.2. Interface Unit RS-232 .....	38

2.4.2.1. Max 232 Sebagai Pengubah Tegangan TTL .....	41
2.5. IC 74HC157A .....	45
2.6. LCD ( Liquid Crystal Display ) M1632 .....	46
2.6.1. Register .....	48
2.7. Buzzer .....	49
2.8. Telepon Seluler .....	50
2.8.1. Nokia 8250 .....	50
2.9. Borland Delphi .....	51
2.9.1. IDE ( <i>Integrated Development Enviroment</i> ) .....	51
2.9.2. Menu Borland Delphi .....	53
2.9.3. Komponen Visual dan Nonvisual .....	54
2.9.4. Komponen <i>TOxygenSMS</i> .....	55
2.9.4.1. <i>Object Inspector TOxygenSMS</i> .....	56
2.9.4.2. Fungsi Internal <i>TOxygenSMS</i> .....	58
<b>BAB III PERANCANGAN DAN PEMBUATAN ALAT</b> .....	61
3.1. Pendahuluan .....	61
3.1.1 Blok Diagram Kcscluruhan Sistem .....	61
3.2. Prinsip Kerja Alat .....	63
3.3. Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	63
3.3.1. RFID ( <i>Radio Frequency Identification</i> ) .....	63
3.3.2. Mikrokontroler AT89S8252 .....	64
3.3.2.1. Perancangan Rangkaian Reset .....	64
3.3.2.2. Perancangan <i>Clock</i> .....	66

3.3.2.3. Perancangan Penggunaan <i>Port</i> pada	
Mikrokontroler AT89S8252 .....	66
3.3.3. Rangkaian Antar Muka RS-232 .....	68
3.3.4. IC 74HC157A .....	69
3.3.5. Perancangan Rangkaian <i>LCD</i> .....	70
3.3.6. Perancangan <i>Interface Ponsel</i> dengan	
<i>PC Notebook ( Laptop )</i> .....	71
Perancangan Perangkat Lunak ( <i>Software</i> ) .....	72
3.4.1. Program Aplikasi Mikrokontroler .....	72
3.4.2. Program Aplikasi Komputer .....	73
3.4.2.1. <i>Component Pallette Delphi7</i> .....	73
3.4.3. FlowChart .....	80
<b>BAB IV PENGUJIAN ALAT</b> .....	82
4.1. Pendahuluan .....	82
4.2. Pengujian RFID .....	82
4.2.1. Tujuan .....	82
4.2.2. Prosedur Pengujian .....	82
4.2.3. Hasil Pengujian .....	85
4.3. Pengujian Mikrokontroler AT89S8252 .....	86
4.3.1. Tujuan .....	86
4.3.2. Prosedur Pengujian .....	86
4.3.3. Hasil Pengujian .....	88
4.3.4. Analisis Hasil Pengujian .....	88



4.4. Pengujian Serial 232 .....	89
4.4.1. Tujuan .....	89
4.4.2. Peralatan Yang Digunakan .....	89
4.4.3. Prosedur Pengujian .....	89
4.4.4. Hasil Pengujian .....	92
4.5. Pengujian Rangkaian Tampilan LCD .....	92
4.5.1. Tujuan .....	92
4.5.2. Peralatan Yang Digunakan .....	92
4.5.3. Prosedur Pengujian .....	92
4.5.4. Hasil Pengujian .....	95
4.6. Pengujian Buzzer .....	95
4.6.1. Tujuan .....	95
4.6.2. Peralatan Yang Digunakan .....	95
4.6.3. Prosedur Pengujian .....	95
4.7. Pengujian Sistem Absensi Berbasis RFID Yang Di informasikan Via SMS .....	97
4.7.1. Tujuan .....	97
4.7.2. Prosedur Pengujian .....	97
4.7.3. Hasil Pengujian .....	97
4.7.4. Proses Identifikasi dan Sistem Absensi .....	98
4.8. Spesifikasi Alat .....	105
<b>BAB V PENUTUP</b> .....	106
5.1. Kesimpulan .....	106

5.2. Saran .....	106
<b>DAFTAR PUSTAKA .....</b>	<b>107</b>
<b>LAMPIRAN-LAMPIRAN</b>	

## DAFTAR GAMBAR

2-1. Konfigurasi ID-12 (RFID Reader).....	8
2-2. <i>Transponder</i> dan <i>Transponder chip</i> .....	10
2-3. Komunikasi antara <i>Reader</i> dan <i>Transponder (Tag)</i> .....	11
2-4. <i>supply</i> dalam <i>transponder</i> .....	12
2-5. Blok diagram model RFID Reader dengan sinyal FSK 125 KHz ....	13
2-6. Blok Diagram Mikrokontroler AT89S8252.....	17
2-7. Skema Pinout AT89S8252 .....	18
2-8. Osilator Eksternal AT89S8252 .....	21
2-9. Koneksi SPI Master dan Slave .....	25
2-10. Format Transfer SPI dengan CPHA = 0.....	26
2-11. Format Transfer SPI dengan CPHA = 1 .....	26
2-12. Timer 2 pada Mode <i>Capture</i> .....	32
2-13. Timer 2 pada Mode <i>Auto Reload</i> DCEN=0 (atas), DCEN=1 (bawah) .....	33
2-14. Timer 2 sebagai <i>Baud Rate Generator</i> .....	35
2-15. Timer 2 dalam <i>Clock Out Mode</i> .....	36
2-16. IC MAX 232 .....	42
2-17. <i>1 byte of Async Data</i> .....	43
2-18. Konektor DB-9 .....	43
2-19. Logic Diagram .....	45
2-20. IC 74HC157 .....	45

2-21. Pin Out LCD M1632.....	47
2-22. Rangkaian <i>Driver Buzzer</i> .....	50
2-23. IDE (Integrated Development Enviroment) .....	52
2-24. <i>Object Inspector TOxygenSMS</i> .....	56
3-1. Diagram Blok Keseluruhan Sistem.....	61
3-2. Rangkaian <i>RFID Reader</i> .....	64
3-3. Perancangan Rangkaian <i>Reset</i> .....	65
3-4. Perancangan Rangkaian <i>Clock</i> .....	66
3-5. Rangkaian MCU AT89S8252 .....	67
3-6. Rangkaian Antar Muka RS-232 .....	69
3-7. Rangkaian IC 74HC157.....	70
3-8. Perancangan Rangkaian Liquid Crystal Display ( LCD ) .....	71
3-9. Interface <i>Notebook</i> dengan <i>Ponsel</i> .....	71
3-10. Diagram Alir Program Mikrokontroler .....	80
3-11. Diagram Alir Program Komputer .....	81
4-1. Kotak Dialog Connection Description .....	83
4-2. Kotak Dialog Connect To .....	83
4-3. Kotak Dialog COM 1 Properties .....	84
4-4. Identifikasi Reader Terhadap kartu .....	84
4-5. Jarak Pembacaan Reader Terhadap kartu .....	86
4-6. Diagram Blok Pengujian Mikrokontroler .....	86
4-7. Hasil Pengujian Tampilan Led .....	88
4-8. Rangkaian Pengujian Transfer Data .....	89

4-9. Diagram Blok Pengujian Rangkaian LCD .....	92
4-10. Hasil Pengujian Tampilan LCD .....	95
4-11. Blok Rangkaian <i>Buzzer</i> .....	96
4-12. Rangkaian Driver <i>Buzzer</i> dan Led .....	96
4-13. Tampilan LCD Koneksi ke PC Harap Tunggu .....	98
4-14. Tampilan LCD Komunikasi ke PC Terkoneksi .....	98
4-15. Tampilan LCD Silahkan Absen .....	99
4-16. Tampilan Layar Monitor Silahkan Absen .....	99
4-17. Tampilan LCD Memproses Data Harap Tunggu .....	99
4-18. Tampilan LCD Absensi Sukses .....	100
4-19. Tampilan Layar Monitor Foto, Nama, Jam Masuk .....	100
4-20. Tampilan Pesan Informasi Siswa Masuk Sekolah .....	101
4-21. Tampilan LCD Data Tidak Terdaftar .....	101
4-22. Tampilan Monitor Kartu Tidak Terdaftar .....	102
4-23. Mengirim Laporan Untuk Siswa Tidak Masuk .....	102
4-24. Tampilan Pesan Informasi Siswa Tidak Masuk Sekolah .....	103
4-25. Tampilan Melihat Laporan sesuai Tanggal .....	103
4-26. Tampilan Mencari Informasi Siswa .....	104
4-27. Tampilan Preview Hasil Laporan .....	104
4-28. Alat Absensi .....	105
4-29. Alat keseluruhan .....	105

## DAFTAR TABEL

2-1. Fungsi Pin & Format Data .....	9
2-2. Fungsi Port 1 .....	19
2-3. Fungsi Khusus Pada Port 3 .....	20
2-4. Mode Operasi Timer 2 .....	31
2-5. Alamat Sumber Interupsi .....	37
2-6. Nama Pin dan Keterangan dari <i>Port</i> RS-23 .....	39
2-7. Fungsi Pin RS-232 dalam DB-9 .....	44
2-8. Kebenaran IC 74HC157 .....	46
2-9. Fungsi Pin – Pin LCD .....	48
4-1. Hasil Pengujian Pembacaan RFID.....	85
4-2. Hasil Pengujian Sistem Mikrokontroler .....	88
4-3. Hasil Pengujian Komunikasi serial .....	92

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Sehubungan dengan perkembangan pola pikir manusia ilmu pengetahuan dan teknologi ternyata mengalami kemajuan terus menerus. Perbaikan terhadap teknologi yang sudah ada terus dilakukan agar menjadi lebih mudah. Salah satu bidang teknologi yang mengalami perkembangan lebih pesat adalah teknologi elektronika yang tidak terlepas dari tuntutan masyarakat yang terus – menerus berkembang sesuai dengan kondisi dan situasi yang dihadapi.

Seiring dengan perkembangan teknologi yang semakin berkembang, maka pemanfaatan teknologi dapat diterapkan dalam berbagai bidang, salah satunya adalah bidang pendidikan. Sering kita melihat suatu kondisi dimana seorang murid yang ijin kepada orang tuanya untuk berangkat ke sekolah tapi ternyata murid tersebut tidak hadir di sekolah. Dari kondisi tersebut maka antara pihak orang tua murid dan pihak lembaga sekolah diharapkan bisa saling berkomunikasi dengan tanpa melakukan suatu pertemuan. Untuk itu penulis mencoba untuk memberikan solusi dengan memanfaatkan teknologi *Wireless* yaitu *RFID (Radio Frequency Identification)* sebagai kartu absensi murid dan penulis juga mencoba memanfaatkan teknologi *celluler* yang sudah populer didunia yaitu layanan *messaging* yang tak lain adalah *SMS (Short Message Service)* sebagai media informasi kehadiran murid di sekolah kepada masing-masing wali murid.

---

Sistem absensinya yaitu setiap murid yang melakukan absensi harus memiliki *tag RFID*, kemudian dengan bantuan *RFID reader* beserta mikrokontroler AT89S8252, dan *LCD*, *tag* tersebut dapat diidentifikasi. Untuk menampilkan data dari murid tersebut, berdasarkan kode dari *Tag RFID* yang telah teridentifikasi dilakukan dengan bantuan komputer. Kemudian setelah proses absensi maka data kehadiran murid tersebut langsung diinformasikan melalui layanan SMS (*Short Message Service*) kepada orang tua / wali murid.

### 1.2. Rumusan Masalah

Mengacu pada permasalahan yang diuraikan pada latar belakang, maka rumusan masalah ditekankan pada:

1. Bagaimana menggunakan RFID sebagai masukan pada mikrokontroler.
2. Bagaimana membuat sistem absensi murid agar dapat ditampilkan pada sebuah monitor.
3. Bagaimana cara mengirimkan informasi kepada orang tua/ wali murid melalui sms (*short message service*).
4. Bagaimana merancang dan membuat perangkat keras (*hardware*) dan perangkat lunak (*software*) agar sistem bekerja dengan baik.

### 1.3. Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan masalah tersebut dibatasi sebagai berikut :

1. Data berisi Foto Siswa, Nama siswa, No. Induk siswa, Jenis kelamin, Kelas, Jam masuk, No Hp, dan Laporan sms.



2. Mikrokontroler yang digunakan adalah AT89S8252, *RFID reader* yang digunakan ID-12, *LCD* yang digunakan adalah tipe M163, HP yang digunakan Tipe Nokia 8250
3. *PhoneNumber* menggunakan format kode negara internasional indonesia yaitu "+62".
4. Yang diinformasikan hanya data absensi masuk dan tidaknya murid di sekolah, yang berisi nama murid dan jam masuk.
5. Menggunakan RS 232 untuk *interface* antara komputer dengan mikrokontroler menggunakan COM1.

#### 1.4. Tujuan

Tujuan dari penulisan skripsi ini adalah untuk meningkatkan kedisiplinan murid di sekolah dan memudahkan pihak lembaga sekolah untuk menginformasikan data kehadiran murid di sekolah kepada orang tua / wali murid secara langsung, sehingga penulis mencoba merancang dan membuat Sistem Absensi Pada Murid Sekolah Menggunakan RFID Dengan Tampilan *LCD* Berbasis Mikrokontroler AT89S8252 Yang Diinformasikan Ke Wali Murid Via SMS.

#### 1.5. Metodologi Penulisan

Metode yang digunakan dalam penulisan tugas akhir ini adalah

1. Studi Pustaka

Memperoleh data dengan cara membaca dan mempelajari buku *literature* yang berhubungan dengan penyusunan skripsi ini.

## 2. Studi Lapangan

Memperoleh data dengan cara praktek secara langsung untuk menunjang pembuatan alat.

## 3. Pengolahan Data

Mengolah data dengan jalan membuat analisa dan menarik kesimpulan dari hasil pengujian yang ada.

### 1.6. Sistematika Penulisan

Sistematika pembahasan dari skripsi ini terdiri dari pokok pembahasan yang saling berkaitan antara satu dengan lainnya, yaitu :

#### **BAB I    Pendahuluan**

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, metodologi dan sistematika penulisan dari alat yang direncanakan.

#### **BAB II   Landasan Teori**

Pada bab ini dibahas tentang teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

#### **BAB III Perencanaan Dan Pembuatan Alat**

Pada bab ini dibahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*).

#### **BAB IV Pengujian Alat**

Pada bab ini dibahas tentang proses serta hasil dari pengujian alat, yang didasarkan oleh pengukuran-pengukuran yang diperlukan.

---

## **BAB V Penutup**

Pada bab ini akan disampaikan kesimpulan dan saran dari perencanaan dan pembuatan sistem ini.

## BAB II

### TEORI DASAR

#### 2.1. Pendahuluan

Pada bab ini akan dibahas mengenai teori dasar yang berkaitan dengan sistem. Teori dasar ini akan membahas tentang komponen dan peralatan pada alat yang dibuat.

#### 2.2. RFID (*Radio Frequency Identification*)

RFID merupakan sebuah sistem yang mampu mengirimkan identitas secara otomatis dengan menggunakan gelombang radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah devais kecil yang disebut *tag* / *transponder* (*transmitter + responder*). *Tag RFID* akan mengenali diri sendiri ketika mendeteksi sinyal dari devais yang kompatibel, yaitu pembaca RFID (*RFID Reader*) dengan *Range* kisaran pembacaan yang bekerja pada frekuensi 125 KHz.

##### 2.2.1. Komponen *RFID Tag*

- ❖ *RFID Tag* atau *transponder*, yang menampung identifikasi data obyek.
  - ❖ *RFID tag reader* atau *transceiver* yang berfungsi untuk membaca dan menulis data *tag*.
  - ❖ *Server database* yang menyimpan kumpulan *record* isi dari *tag*.
-

### 2.2.1.1. Tag

Tag tersusun dari *microchip* yang berfungsi untuk menyimpan dan sebuah antena *chip* mikro itu sendiri yang ukurannya sekitar 0.4 mm. *Chip* tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung tipe memorinya yaitu *read-only* dan *read-write*.

Klasifikasi *tag* dibedakan menjadi tiga yaitu :

- ❖ *Tag aktif* : mempunyai sumber tenaga seperti baterai dan dapat dilakukan komunikasi untuk dibaca dan ditulis.
- ❖ *Tag semi-pasif* : mempunyai baterai tetapi hanya merespon transmisi yang datang ( *incoming transmissions* ).
- ❖ *Tag pasif* : menerima tenaga dari *reader*, antena yang akan menjadi sumber tenaga dengan memanfaatkan medan magnet yang ditimbulkan dari pembaca (*reader*).

Empat macam frekuensi yang digunakan RFID tag adalah: tag frekuensi rendah (125 atau 134.2 KHz), tag frekuensi tinggi (13.56 MHz), tag UHF (868 sampai 956 MHz) dan tag gelombang mikro (2.45 GHz).

### 2.2.1.2. Tag Reader

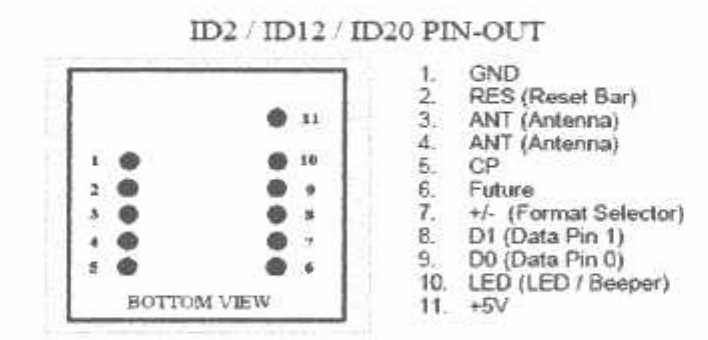
*Tag reader* digunakan untuk membaca data yang ada pada *tag* melewati RF *interface*. Untuk menambah fungsi *reader* dilengkapi dengan *internal storage*, dan aplikasi perangkat lunak untuk menyimpan data pada *server database*. Pada prakteknya *tag reader* dapat berupa perangkat keras yang terletak pada suatu tempat yang tetap. Pada aplikasinya *tag reader* dapat membaca sendiri *tag* yang dideteksi ( *smart self* ). *Tag reader smart self* dapat mendeteksi ketika ada penambahan *tag* yang

keluar. Pada dasarnya *tag reader* merupakan suatu peralatan yang sederhana dan dapat digabungkan kedalam perlengkapan *mobile* seperti telepon.

Saluran (*chanel*) dari *reader* ke *tag* disebut dengan saluran *forward* (*forward chanel*), saluran *tag* ke *reader* disebut dengan saluran *backward* (*backward chanel*).

Spesifikasi Reader ID-12 :

- Power Requirement : 5V@13mA nominal
- Card Format : Temec Q55555 or compatible
- Frequency : 125 KHZ
- Encoding : Manchester 62bit, modulus64
- I/O Output Current : 20mA sink/source
- Drive Current : 300 mA
- Antenna : 100 Volt PKPK



**Gambar 2-1.** Konfigurasi ID-12 (RFID Reader)  
(Sumber : [www.alldatasheet.com](http://www.alldatasheet.com) ID Series DataSheet)

Tabel 2-1. Fungsi Pin & Format Data

Pin No.	Description	ASCII
Pin 1	Zero Volt and Tuning Capacitor Ground	GND 0 V
Pin 2	Strap to +5 Volt	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna
Pin 4	To External Antenna	Antenna
Pin 5	Card Present	No Fuction
Pin 6	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND
Pin 8	Data 1	CMOS
Pin 9	Data 0	TTL Data (Inverted)
Pin 10	3.1 KHz Logic	Beeper / LED
Pin 11	DC Voltage Supply	+5 V

(Sumber : [www.alldatasheet.com](http://www.alldatasheet.com) ID Series DataSheet)

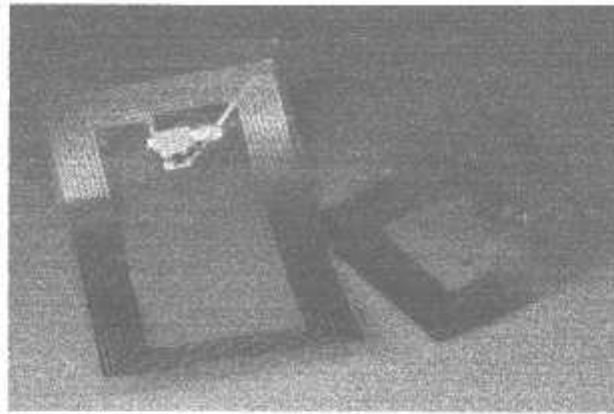
2.2.1.3. Server Database

Untuk menyimpan data yang ada pada tag digunakan server database.

2.2.2. Mekanisme RFID

2.2.2.1. Prinsip Kerja Reader dan Tag/Transpoder

Suatu transponder secara induktif yang tergabungkan terdiri atas suatu data elektronik di dalam suatu microchip yang pada umumnya tunggal dan suatu coil area besar yang berfungsi sebagai suatu antena.



**Gambar 2-2.** *Transponder dan Transponder chip*  
(Sumber : RFID Handbook)

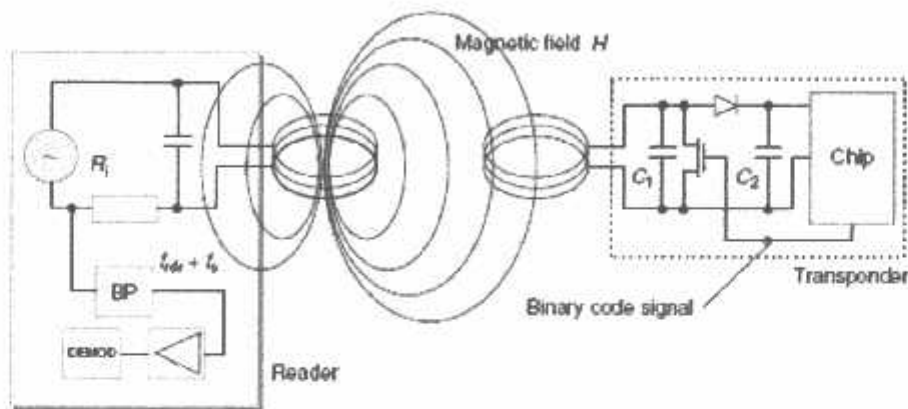
Secara induktif *transponders* dioperasikan dengan *pasif* yaitu semua energi yang diperlukan untuk operasi *microchip* harus disajikan oleh pembaca *reader*. antena pembaca menghasilkan suatu bidang elektro magnet frekuensi tinggi, yang menembus penampang-lintang area *coil* dan area di sekitar *coil* itu. Sebab panjang gelombang cakupan frekuensi menggunakan *low frekuensi* ( 125 kHz – 135 kHz )

Suatu bidang elektro magnet yang dipancarkan menembus *coil* antena *transponder*, yang mana saat terinduksi, suatu tegangan dihasilkan *coil antena transponder*. Tegangan ini berfungsi sebagai power untuk pengaktifan data dalam *microchip*.

Suatu kapasitor *C* yang dihubungkan paralel dengan *coil* antena pembaca berkombinasi dengan induksi *coil* antena untuk membentuk suatu rangkaian resonansi paralel, dengan suatu frekuensi resonan yang bersesuaian dengan frekuensi transmisi pembaca yang dihasilkan di dalam *coil* antena pembaca akan meningkatkan rangkaian resonan yang paralel tersebut, yang dapat digunakan untuk menghasilkan kekuatan bidang elektro magnet.



*Coil* antenna *transponder* dan kapasitor untuk membentuk suatu rangkaian resonan dan mengatur kesesuaian kepada frekuensi transmisi pembaca. Tegangan di *transponder coil* akan meningkatkan rangkaian resonan paralel pada *tag*.



**Gambar 2-3.** Komunikasi antara *Reader* dan *Transponder* (*Tag*).

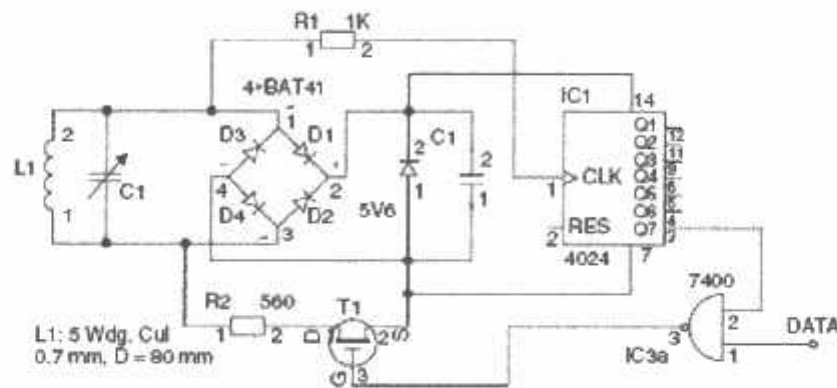
(Sumber : *RFID Handbook*)

#### 2.2.2.2. Pembacaan

Jika suatu resonan transponder ( yaitu *self-resonant* frekuensi *transponder* bersesuaian dengan frekuensi transmisi pembaca ) berada di dalam bidang magnetis antenna pembaca akan mendapatkan energi seri dari medan magnet itu. Konsumsi energi terjadi ketika adanya penurunan-voltase di dalam antenna pembaca.

*Switch* transponder sekali-sekali melakukan pembalasan beban di antenna *transponders* dan itu mempengaruhi perubahan tegangan di antenna pembaca juga mempunyai efek dari suatu modulasi amplitudo tegangan oleh antenna *transponder* saat *menswitch* sekali-kali resistor beban dikendalikan oleh data, kemudian data ini dapat *ditransfer* dari *transponder* kepada pembaca. Perpindahan data jenis ini disebut modulasi beban.

Untuk mengambil kembali data di dalam pembaca, pengaturan voltase di penyearah antena pembaca digunakan demodulasi untuk membangkitkan modulasi sinyal.



**Gambar 2-4.** *supply dalam transponder*

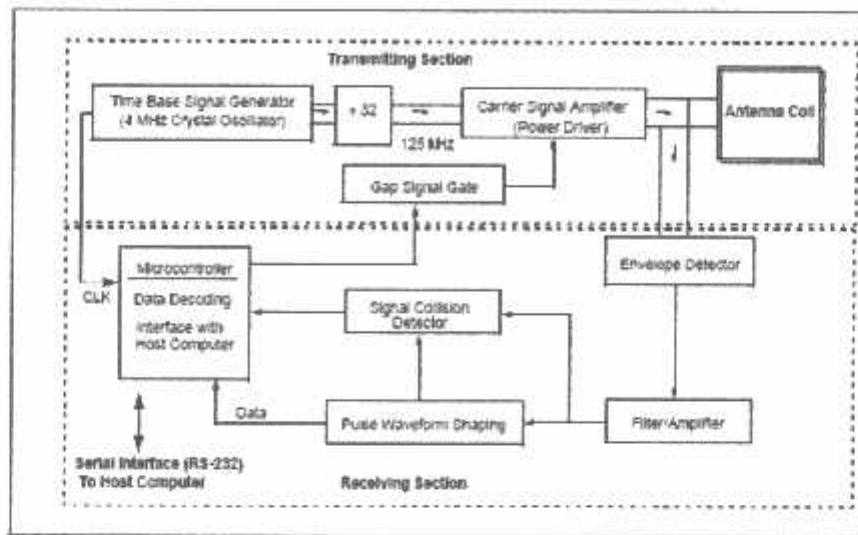
(*source : RFID Handbook*)

Gambar di atas, jika resistor beban di dalam *transponder* diswitch *on/off* pada suatu frekuensi dasar tinggi  $f_H$ , kemudian dua garis spektrum diciptakan jauh  $f_H$  di sekitar frekuensi pembaca, dan ini dapat dengan mudah dideteksi (  $f_H$  harus kurang dari  $f_{Reader}$  ). Di dalam istilah teknologi radio frekuensi dasar yang baru dapat disebut *subcarrier*. Perpindahan data dengan FSK atau PSK modulasi *subcarrier* pada saat data mengalir akan menghadirkan suatu modulasi amplitudo *subcarrier*.

#### 2.2.2.3. Modulasi Yang Digunakan

Pembaca terdiri dari suatu pemancaran dan suatu penerima bagian. Bagian pemancaran meliputi suatu generator frekuensi *carrier*, gerbang gap sinyal, dan suatu rangkaian antenna. Bagian penerima meliputi suatu detektor puncak gelombang, suatu *amplifier/filter* sinyal, suatu detektor *signal-collision*, dan suatu *microcontroller* untuk pengolahan data.

Pembacaan juga berkomunikasi dengan suatu komputer eksternal. Suatu diagram blok dasar RFID pembaca yang ditunjukkan sebagai berikut:



**Gambar 2-5.**

Blok diagram model RFID Reader dengan sinyal FSK 125 KHz.

(sumber : *MicroID 125 KHz RFID System Design Guide*)

125-KHz sinyal *carrier* dihasilkan dari 4-MHz *timebase* sinyal generator dan suatu osilator kristal. Sinyal 125-KHz dikuatkan ke dalam *NOR gate* dan *two-stage* menggerakkan *amplifier* sinyal *carrier* 125 KHz kemudian diteruskan ke dalam rangkaian antenna yang dibentuk oleh L ( 162 mH ) dan C ( 0.01 mF ), rangkaian L dan C membentuk suatu *series-resonant* dengan suatu 125-KHz frekuensi resonan, sinyal *carrier* ( 125 KHz ) disaring untuk dikirim ke *coil* antenna. Rangkaian tersebut menyediakan suatu impedansi minimum di frekuensi resonan. Ini mengakibatkan memaksimalkan di frekuensi yang resonan.

Sinyal Gap mengendalikan 125-kHz rangkaian pengarah antenna sepanjang isyarat gap tinggi maka tidak ada RF sinyal *coil* antenna selama periode gap ini.

2.2.2.4. Pengiriman Data

Saat model alat identifikasi sangatlah bermacam-macam, ada yang berupa kartu dengan lubang, *barcode*, *RFID*, dll. *RFID* ( *RF Identification* ) merupakan suatu alat untuk identifikasi yang biasanya ditempelkan pada barang atau dibuat menjadi kartu. Pembacaan fromat data yang dikeluarkan oleh *RFID reader* dengan format *output* ASCII.

*RFID reader* mempunyai banyak sekali tipe, antara lain : ID-10, ID-12, EM-13, dll. Biasanya *RFID reader* ini memiliki dua bentuk *output serial* yaitu : ASCII dan *Wiegand 26-bit*. Yang sering digunakan adalah *output* dengan format ASCII, karena *output* ini sangat mudah untuk dihubungkan pada mikrokontroler atau *PC* menggunakan komunikasi serial UART.

2.2.2.5. Format Pembacaan ASCII

Output yang memiliki format *ASCII* memiliki struktur sebagai berikut :

02	10 Data Karakter ASCII	2 Karakter ASCII (Checksum )	CR	LF	03
----	---------------------------	------------------------------	----	----	----

(Sumber : [www.adilam.com.au/RFID/rfid.html](http://www.adilam.com.au/RFID/rfid.html))

Checksum merupakan hasil EXOR (*Exclusive OR*) dari 5 biner data *byte*, misalnya data *output serial* ( dalam *hexadesimal* ) yang kita tangkap adalah sebagai berikut :

02	30	34	36	32	30	31	44	37	36	43	44	43	0D	0A	03
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Langkah pertama adalah merubah semua nilai data diatas menjadi karakter *ASCII*. Misalnya 30H menjadi karakter “0”, 34H menjadi karakter “4” dst. Langkah

kedua adalah menyusun data - data tersebut kedalam format data *ASCII*. Dari contoh data *hexadesimal* diatas maka dapat dibuat tabel seperti tabel dibawah ini :

Data Heksa	30	34	36	32	30	31	44	37	36	43
Data ASCII	0	4	6	2	0	1	D	7	6	C

Untuk data yang pertama yaitu angka “04” merupakan data untuk jenis – jenis kartu. Yang digunakan adalah data ke 3 s/d 10. Hasil konversi dari data heksa ke dalam data ASCII adalah “ 6201D76C “. Kemudian gabungkan data karakter *ASCII* menjadi bilangan *hexadesimal*, setelah itu konversikan bilangan *hexadesimal* tersebut ke desimal. Hasilnya adalah 6201D76C Heksa menjadi 1644287852, angka – angka ini merupakan no kartu sebenarnya yang tertera pada badan kartu yang disebut *tag RFID*.

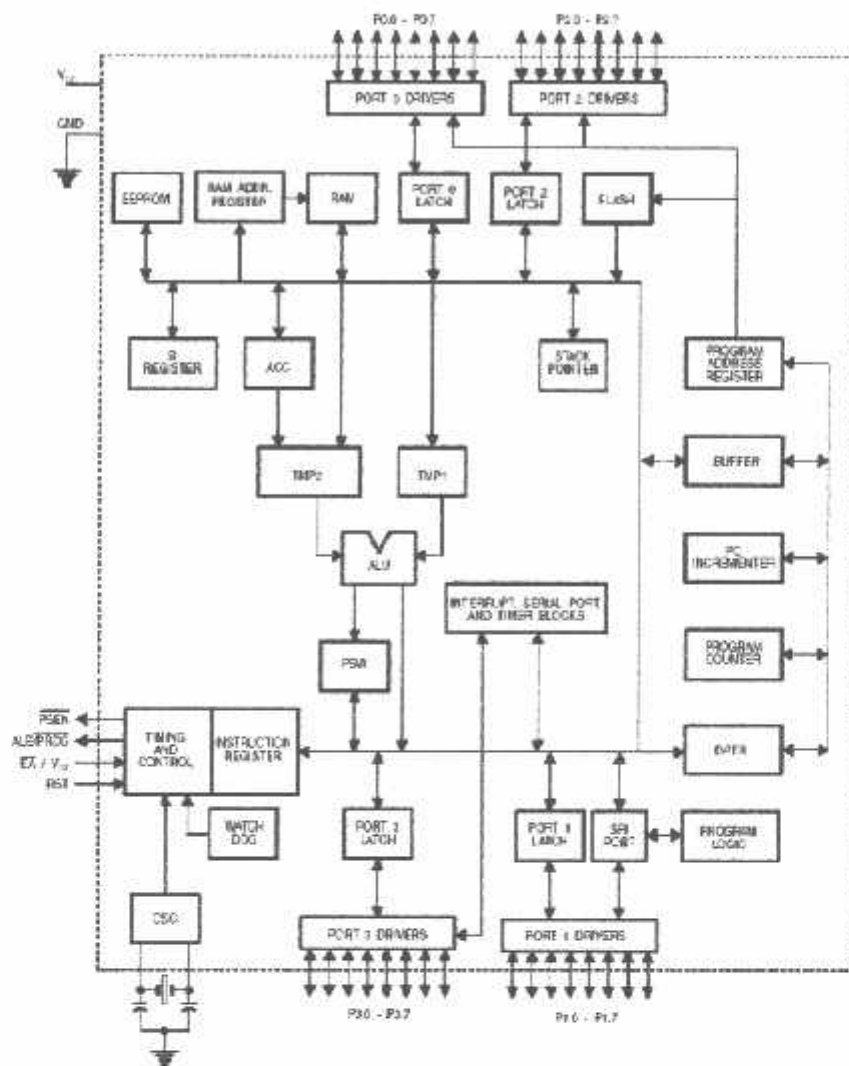
### 2.3. Mikrokontroler ATMEL AT89S8252

Mikrokontroler Atmel AT89S8252 merupakan pengembangan dari mikrokontroler standar MCS-51. Hal-hal yang terdapat pada penjelasan mikrokontroler MCS-51 juga berlaku untuk mikrokontroler Atmel AT89S8252. Hanya mikrokontroler Atmel AT89S8252 memiliki beberapa kelebihan bila dibandingkan dengan mikrokontroler MCS-51. Karena adanya fasilitas tambahan yang tidak terdapat pada mikrokontroler MCS-51, maka mikrokontroler Atmel AT89S8252 dapat menggantikan mikrokontroler MCS-51.

#### 2.3.1. Fasilitas Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 memiliki berbagai fasilitas penting yang dimiliki oleh AT89S8252 sebagai sebuah pengendali, antara lain adalah sebagai berikut :

- ❖ Memiliki 8 Kilobytes Flash PEROM (*Programmable and Erasable Read Only Memory*) yang dapat diprogram ulang dengan fasilitas SPI (*Serial Programming Interface*). Memori Flash PEROM ini dapat bertahan untuk dihapus dan ditulis ulang kira-kira sebanyak 1000 kali.
- ❖ Memiliki 2 Kilobytes EEPROM yang dapat berfungsi sebagai data memori. EEPROM ini dapat bertahan untuk dihapus dan ditulis ulang sekitar 100.000 kali.
- ❖ Beroperasi pada kisaran tegangan 4V sampai dengan 6V.
- ❖ Mendukung pewaktuan 0 Hz sampai 24 MHz *clock speed*.
- ❖ Fasilitas pengamanan program memori sebanyak tiga level pengamanan data (*program memory lock*).
- ❖ Memiliki RAM internal sebanyak 256 byte
- ❖ Memiliki 32 unit input dan output (*port pins*) yang dapat diprogram sebagai jalur masukan dan keluaran. Terbagi dalam 4 port paralel dan 1 port serial yang dapat berjalan dengan mode dua arah (*full duplex*).
- ❖ Memiliki fasilitas *Timer* dan *Counter* sebanyak tiga buah.
- ❖ Memiliki fasilitas *watchdog timer* yang dapat diprogram sesuai keinginan.
- ❖ Memiliki dua buah DPTR (*Data Pointer*). Data pointer berguna untuk pengalamatan memori dengan alamat 16 bit.



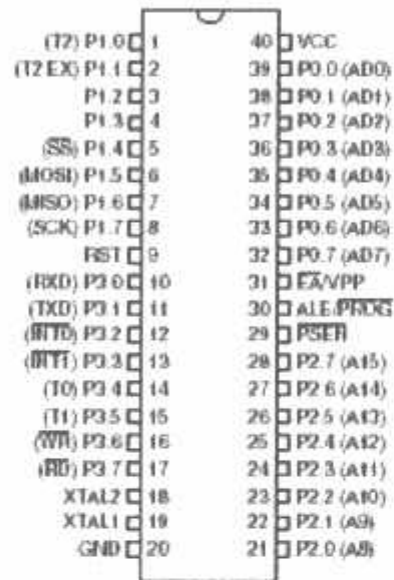
**Gambar 2-6.** Blok Diagram Mikrokontroler AT89S8252

(Sumber : Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

Seperti halnya mikrokontroler yang lainnya, mikrokontroler AT89S8252 memiliki dimensi yang cukup kecil. Dimensinya yang cukup kecil membuat mikrokontroler sangat berguna dalam perancangan *embedded control application*. Berikut ini adalah penjelasan dari dimensi jalur-jalur penting dalam dimensi mikrokontroler, yang juga dikenal dengan nama *pin*.

### 2.3.2. Pinout Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 terdiri dari 40 pin dengan pinout sebagai berikut:



**Gambar 2-7. Skema Pinout AT89S8252**

(Sumber : Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

Fungsi-fungsi tiap pinnya adalah sebagai berikut :

- ❖ VCC (Supply tegangan), pin 40
- ❖ GND (Ground) , pin 20
- ❖ Port 0, pin 32-39

Merupakan port input-output dua arah, tanpa internal pull-up dan konfigurasi sebagai multipleks bus alamat rendah (A0-A7) dan data selain pengaksesan program memory dan data memory eksternal.

- ❖ Port 1, pin 1-8

Merupakan port input-output dua arah dengan *internal pull-up*.



Fungsi khusus pin-pin pada port1 sebagai berikut :

Tabel 2-2. Fungsi Port 1

Nama Pin	Kegunaan Khusus
P1.0	T2 ( <i>input timer</i> atau <i>counter 2</i> dari sumber <i>clock</i> eksternal)
P1.1	T2EX ( <i>Timer/Counter 2 capture/reload trigger</i> dan <i>direction control</i> )
P1.4	SS (untuk memilih <i>slave port</i> )
P1.5	MOSI ( <i>Master data output</i> , masukan <i>slave data input</i> pin untuk fasilitas SPI)
P1.6	MISO ( <i>Master data input</i> , keluaran <i>slave data output</i> pin untuk fasilitas SPI)
P1.7	SCK ( <i>Master clock output</i> , masukan untuk pin <i>slave clock</i> untuk fasilitas SPI channel)

(Sumber :DataSheet AT89S8252 [www.atmel.com](http://www.atmel.com))

❖ Port 2, pin 21-28

Merupakan port input-output dengan *internal pull-up*. Mengeluarkan alamat tinggi selama pengambilan program memori eksternal.

❖ Port 3, pin 10-17

Merupakan port input-output dengan *internal pull-up*, dimana port 3 juga memiliki fungsi khusus dan dapat dilihat pada table berikut :

**Tabel 2-3.** Fungsi Khusus Pada Port 3

Nama Pin	Kegunaan Khusus
P3.0	RXD (port penerima komunikasi serial)
P3.1	TXD (port pengirim komunikasi serial)
P3.2	$\overline{INT0}$ (masukan untuk fasilitas interupsi eksternal 0)
P3.3	$\overline{INT1}$ (masukan untuk fasilitas interupsi eksternal 1)
P3.4	T0 (masukan untuk fasilitas timer eksternal 0)
P3.5	T1 (masukan untuk fasilitas timer eksternal 1)
P3.6	$\overline{WR}$ (sinyal tulis memori data eksternal)
P3.7	$\overline{RD}$ (sinyal baca memori data eksternal)

(Sumber :Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

❖ RST (Reset), pin 9

Input reset merupakan reset master untuk AT89S8252

❖ ALE /Prog (*Address Latch Enable*), pin 30

Digunakan untuk menahan alamat memori eksternal selama pelaksanaan intruksi.

❖ PSEN (*Program Store Enable*), pin 29

Merupakan sinyal pengontrol yang memperbolehkan program memori eksternal masuk ke dalam bus.

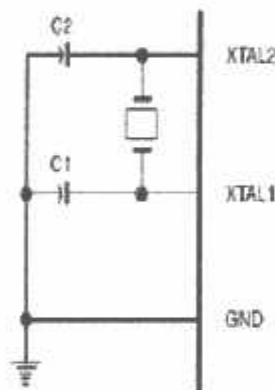
❖ EA/VPP (*External Access*), pin 31

Dapat diberikan logika rendah (Ground) atau logika tinggi (+5 Volt). Jika diberikan logika tinggi maka mikrokontroler akan mengakses program dari

ROM internal (EEPROM/Flash Memori), dan jika diberikan logika rendah maka mikrokontroler akan mengakses program dari memori eksternal.

❖ X-TAL 1 dan X-TAL 2, pin 18, 19

Pin ini dihubungkan dengan kristal bila menggunakan osilator internal. X-TAL 1 merupakan masukan ke rangkaian osilator internal sedangkan X-TAL 2 keluaran dari rangkaian osilator internal. Untuk keperluan ini diperlukan kapasitor penstabil sebesar 30 pF. Dan nilai dari X-TAL tersebut antara 4-24Mhz. untuk lebih jelasnya dapat dilihat gambar pemasangan X-TAL serta kapasitor yang digunakannya.



**Gambar 2-8. Osilator Eksternal AT89S8252**

(Sumber :Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

### 2.3.3. SFR tambahan pada mikrokontroler Atmel AT89S8252

Selain memiliki SFR seperti halnya pada mikrokontroler MCS-51, mikrokontroler Atmel AT89S8252 memiliki tambahan SFR. Hal ini tak lain adalah karena terdapatnya tambahan fitur pada mikrokontroler ini. Jadi SFR tambahan ini adalah SFR untuk mengontrol alat tambahan pada mikrokontroler Atmel AT89S8252.

SFR tambahan ini meliputi:

- ❖ T2CON (*Timer 2 Register*), dengan alamat 0C8H.
- ❖ T2MOD (*Timer 2 Mode*), dengan alamat 0C9H.
- ❖ WMCON (*Watchdog and Memory Control Register*), dengan alamat 96H.
- ❖ SPCR (*SPI Control Register*), dengan alamat D5H.
- ❖ SPSR (*SPI Status Register*), dengan alamat AAH.
- ❖ SPDR (*SPI Data Register*), dengan alamat 86H.

2.3.3.1. SFR untuk Timer 2

Pada mikrokontroler Atmel AT89S8252 terdapat tambahan sebuah *Timer/Counter* yang diberi nama *Timer 2* (sehingga mikrokontroler Atmel AT89S8252 memiliki 3 *Timer/Counter* yaitu *Timer/Counter 0*, *Timer/Counter 1*, dan *Timer/Counter 2*).

Pada *Timer/Counter 2* ini dikendalikan oleh *special function register* yang bernama T2CON (*Timer 2 Control*), T2MOD (*Timer 2 Mode*), dan sepasang register RCAP2H, RCAP2L merupakan register *capture/reload* untuk *Timer 2* dalam 16 bit *capture mode* atau *auto reload mode*.

Register T2CON yang beralamat di 0C8H memiliki bit-bit sebagai berikut:

MSB				LSB			
TF2	EXF2	RCLK	TCL	KEXEN2	TR2	C/T2	CP/RL2
Bit		Keterangan					
TF2		Bendera <i>overflow Timer 2</i> , diset oleh <i>Timer 2</i> dan harus di-clear lewat <i>software</i>					

- EXF2** Bendera luar *Timer 2* diset saat suatu *capture* atau *reload* disebabkan oleh transisi negatif pada T2EX dan EXEN2=1. Jika sela *Timer 2* diaktifkan, EXF2 =1 akan menyebabkan CPU mencabang ke rutin sela *Timer 2*. EXF2 harus di-clear dengan *software*. EXF2 tidak menyebabkan sela pada *mode up/down counter* (DCEN=1).
- RCLK** *Receive clock enable*. Jika diset menyebabkan serial port menggunakan pulsa *overflow Timer 2* sebagai detak penerimaan pada serial port untuk mode 1 dan 3. jika RCLK=0 menyebabkan pulsa *overflow Timer 1* yang digunakan sebagai detak.
- TCLK** *Transmit clock enable*. Jia diset menyebabkan serial port menggunakan pulsa *overflow Timer 2* sebagai detak pengiriman. Jika TCLK=0 menyebabkan pulsa *overflow Timer 1* yang digunakan sebagai detak pengiriman.
- EXEN2** *Timer 2 eksternal enable*. Jika diset memungkinkan *capture* atau *reload* terjadi sebagai hasil dari transisi negative pada pin T2EX jika *Timer 2* tidak sedang digunakan sebagai *baud rate generator* untuk serial port. Jika EXEN2=0 menyebabkan *Timer 2* akan melakukan apa-apa kejadian pada pin T2EX.
- TR2** Bit untuk mengatur *start/stop* untuk *Timer 2*. Jika TR2=1 *Timer* akan aktif.
- C/T2** Bit pemilihan *Timer* atau *Counter* untuk *Timer 2*. Jika C/T2=0 maka terpilih fungsi *timer*, C/T2=1 untuk fungsi *counter*.

**CP/RL2** Pemilihan *Capture/Reload*. Jika diset maka proses *capture* akan terjadi pada transisi negatif pada pin T2EX jika bit EXEN2=1. Jika bit ini di-clear maka proses *reload* otomatis akan terjadi saat *Timer 2* overflow atau transisi negative terjadi pada pin T2EX saat bit EXEN2=1. Jika bit RCLK atau TCLK diset maka bit ini menjadi tidak diprhitungkan (*ignore*). Hal ini karena *Timer 2* dipakai sebagai *baud rate generator* pada serial port.

SFR ini memiliki nilai pada saat reset : 0000 0000B.

*Timer 2* juga memiliki SFR yang bernama T2MOD (*Timer 2 Mode Control Register*) yang beralamat di 0C9H dan memiliki nilai pada saat reset XXXX XX00B.

Bit-bit pada T2MOD adalah sebagai berikut :

MSB						LSB	
-	-	-	-	-	-	T2OE	DCEN
Bit	Keterangan						
T2OE	Timer 2 Output Enable bit						
DCEN	Jika diset memungkinkan Timer/Counter sebagai up/down counter.						

2.3.3.2. SFR Pengontrol SPI

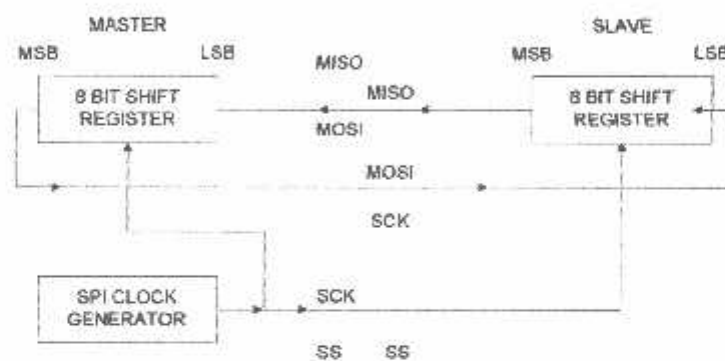
AT89S8252 memiliki fasilitas SPI (*Serial Peripheral Interface*) yang memungkinkan transfer data kecepatan tinggi secara sinkron mikrokontroler Atmel AT89S8252 dengan peripheral atau antar AT89S8252.

Fitur SPI pada AT89S8252 meliputi :

1. *Full Duplex*. 3 kawat dengan transfer data secara sinkron
2. Operasi Master atau Slave
3. Frekuensi maksimum 6 MHz

4. Sistem data transfer MSB dahulu atau LSB dahulu
5. 4 bit rate terprogram
6. Bendera sela pada akhir transmisi
7. *Write Collision Flag Protection*
8. Bangun dari mode *idle* (hanya untuk mode slave)

Gambar berikut menunjukkan hubungan antara CPU master dan slave :



**Gambar 2-9.** Koneksi SPI Master dan Slave

(Sumber : Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

Pin SCK adalah keluaran detak pada mode master, tetapi merupakan detak masukan pada mode slave. Menulis ke SPI data register pada CPU master akan memulai SPI *clock generator*, dan data yang ditulis digeser keluar pada pin MOSI dan menuju pin MOSI pada CPU slave.

Setelah menggeser 1 byte, SPI *clock generator* akan berhenti, dan akan mengaktifkan bendera (*flag*) selesai pengiriman (SPIF). Jika kedua bit pengaktif serial port (ES) di-set, suatu sela akan dibutuhkan.

Bit pemilih *slave select* (SS) pada port 1 pin 4 (P1.4) dibuat rendah untuk memilih suatu alat SPI individual sebagai slave. Jika pena ini tinggi, maka port SPI tidak diaktifkan dan pin MOSI (P1.6) dapat digunakan sebagai masukan.





- SPE** Bit pengaktif SPI. Jika  $SPE=1$  mengaktifkan kanal SPI dan menghubungkan SS, MOSI, MISO, dan SCK ke pin P1.4, P1.5, P1.6, dan P1.7. Jika  $SPE=0$  akan melumpuhkan kanal SPI.
- DORD** Data Order. Jika  $DORD=1$  akan memilih LSB terlebih dahulu untuk dikirimkan. Jika  $DORD=0$  akan memilih MSB terlebih dahulu untuk dikirimkan.
- MSTR** Bit pemilih fungsi *Master* atau *Slave*. Jika  $MSTR=1$  akan memilih mode Master. Jika  $MSTR=0$  akan memilih mode Slave.
- CPOL** Polaritas detak. Jika  $CPOL=1$ , SCK akan tinggi saat kondisi idle. Jika  $CPOL=0$ , SCK pada alat Master akan rendah jika tidak sedang mengirimkan data.
- CPHA** Fasa detak. Bit ini bersama dengan bit CPOL mengontrol hubungan antara detak dengan data antara Master dan Slave.
- SPR1** SPI Clock Rate Select. Kedua bit ini mengontrol rate SCK pada alat yang dikonfigurasi sebagai Master.
- SPR0** SPR1 dan SPR0 tidak berefek pada slave. Hubungan antara SPR1, SPR0, dan SCK adalah sebagai berikut :

SPR1	SPR0	SCK=F osc dibagi dengan
0	0	4
0	1	16
1	0	64
1	1	128

Pada saat reset, register ini akan bernilai 0000 01XXB. (X=*don't care*)

Register SPSR (SPI Status Register) yang beralamat di AAH dan memiliki nilai reset 0000 0000B mempunyai bit-bit sebagai berikut :

MSB								LSB
SPIF	WCOL	-	-	-	-	-	-	-
Bit	Keterangan							
SPIF	Bendera sela SPI. Jika suatu pengiriman secara serial telah selesai, bit SPIF akan di-set dan suatu sela akan dibangkitkan jika bit SPIE=1 dan bit ES=1. Bit SPIF di-clear dengan membaca register status SPI dengan SPIF dan WCOL di-set, dan kemudian mengakses data register SPI.							
WCOL	Write Collision Flag. Bit WCOL akan di-set jika SPI data register sedang menulis saat transfer data. Selama transfer data, hasil pembacaan register SPDR mungkin akan salah, dan penulisan kepadanya tidak berefek.							

Register terakhir adalah register SPDR (SPI Data Register) dengan alamat 86H dan memiliki nilai pada saat reset yang tidak berubah. Bit-bit pada register ini adalah sebagai berikut :

MSB							LSB
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0

AT89S8252 mempunyai dua buah *Power-Saving Mode* yang dapat diatur melalui *software*, yaitu: IDE Mode yang akan menghentikan CPU sebagai RAM, *Timer/Counter*, *Serial Port* dan *Interrupt system* yang tetap berfungsi. *Power Down Mode* ini yang akan menyimpan di RAM dan akan menahan *Oscillator* untuk tidak mengaktifkan *chip* yang lain sampai terjadi reset secara *hardware*.

#### 2.3.4. Organisasi Memori

Organisasi memori pada mikrokontroler AT89S8252 dapat dibagi menjadi dua bagian besar yaitu memori program dan memori data. Pembagian tersebut didasarkan atas fungsi dari penyimpanan data maupun program. Memori program digunakan untuk menyimpan instruksi-instruksi yang akan dijalankan oleh mikrokontroler, sedangkan memori data digunakan sebagai tempat yang sedang diolah mikrokontroler.

Program mikrokontroler disimpan dalam memori program berupa ROM. Mikrokontroler AT89S8252 dilengkapi dengan ROM internal, sehingga untuk menyimpan program tidak digunakan ROM eksternal yang terpisah dari mikrokontroler. Agar tidak menggunakan memori program eksternal, penyemat  $\overline{EA}$  dihubungkan dengan Vcc (logika 1).

Memori program mikrokontroler menggunakan alamat 16 Bit mulai dari 0000H - 0FFFFH sehingga kapasitas penyimpanan program maksimal adalah 4Kbyte. Sinyal  $\overline{PSEN}$  (*Program Store Enable*) tidak digunakan jika menggunakan memori program internal.

Selain program mikrokontroler AT89S8252 juga memiliki data internal sebesar 128 byte dan mampu mengakses memori data eksternal sebesar 64 Kbyte. Semua memori data internal dapat dialamati dengan data langsung atau tidak langsung. Ciri dari pengalamatan langsung adalah *operand* adalah alamat register yang berisi alamat data yang akan diolah. Sebagian memori tersebut dapat dialamati dengan pengalamatan register, dan sebagian lagi dapat dialamati dengan memori satu

---

bit. Untuk membaca data digunakan sinyal  $\overline{RD}$ , sedangkan untuk menulis digunakan sinyal  $\overline{WR}$ .

### 2.3.5. Timer 2

Pada mikrokontroler Atmel AT89S8252 terdapat tambahan Timer 2. *Timer/Counter* dapat digunakan sebagai *generator baud rate* untuk serial port. Pada standar MCS-51 biasanya digunakan Timer 1 sebagai penghasil baud rate. Pada AT89S8252 selain dapat menggunakan Timer 1 sebagai penghasil baud rate (untuk menjaga kompatibilitas dengan MC-51) juga dapat menggunakan Timer 2 sebagai penghasil baud rate untuk serial port.

Timer 2 ini merupakan *Timer/Counter* yang berukuran 16 bit yang dapat beroperasi sebagai Timer (dengan detak dari sistem detak mikrokontroler) atau dapat beroperasi sebagai penghitung kejadian (*event counter*) dengan detak dari luar. Untuk mengatur fungsi ini dilakukan dengan mengatur bit C/T2 pada SFR T2CON. Jika bit ini tinggi maka akan terpilih fungsi counter (C), tetapi jika bit ini rendah maka akan terpilih fungsi Timer 2.

Timer 2 ini memiliki 3 mode operasi, yaitu *capture*, *auto reload (up dan down counting)*, dan *baud rate generator*. Untuk memilih mode ini dilakukan dengan mengatur bit pada SFR T2CON (Timer 2 Control Register).

Timer 2 terdiri dari 2 buah timer 8 bit register yaitu TH2 dan TL2. Pada fungsi timer, register TL2 dinaikkan (*increment*) tiap siklus mesin. Karena siklus mesin terdiri dari 12 periode osilasi, maka *count rate* menjadi  $1/12$  dari frekuensi osilator.

Pada fungsi counter, register dinaikkan berdasarkan tanggapan adanya transisi tinggi ke rendah pada pena yang bersesuaian (dalam hal ini pin T2 atau P1.0). Pada fungsi ini, masukan luar akan disampling selama S5P2 dari tiap siklus mesin. Jika hasil *sampling* menunjukkan logika tinggi pada selama satu siklus dan rendah pada siklus selanjutnya, maka akan terdeteksi transisi dari tinggi ke rendah dan akibatnya penghitungan akan dinaikkan. Nilai penghitungan yang baru akan muncul pada register selama S3P1 dari siklus setelah transisi tinggi ke rendah terdeteksi.

Tabel berikut menunjukkan mode operasi yang dapat dijalankan Timer 2 :

Tabel 2-4. Mode Operasi Timer 2

RCLK+TCLK	CP/RL2	TR2	MODE
0	0	1	16 bit <i>auto reload</i>
0	1	1	16 bit <i>capture</i>
1	X	1	<i>baud rate generator</i>
X	X	0	off

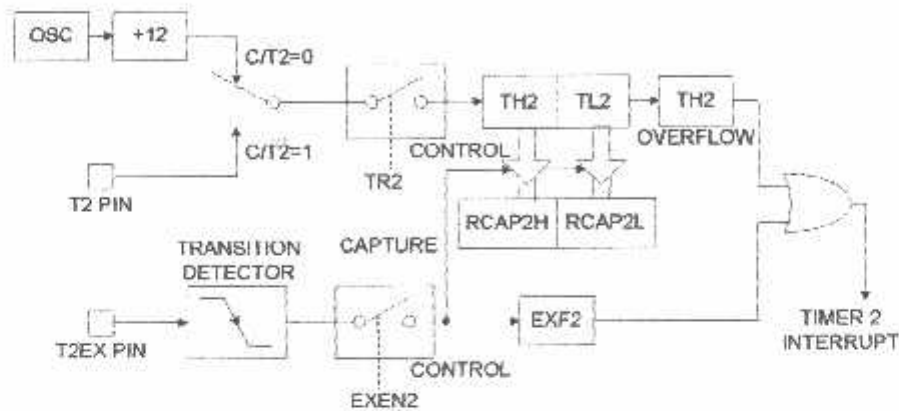
(Sumber :Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com) )

2.3.5.1 Mode Capture

Pada mode ini dua pilihan dipilih oleh bit EXEN2 pada SFR T2CON. Jika EXEN2=0, Timer 2 merupakan 16 bit timer atau counter yang jika telah overflow (melimpah) akan mengeset bit TF2 pada T2CON.

Bit ini kemudian dapat digunakan untuk menghasilkan sela. Jika EXEN2=1, Timer 2 akan berlaku sama, tetapi suatu transisi tinggi ke rendah (1 to 0) pada pin T2EX (P1.1) akan menyebabkan nilai sekarang pada TH2 dan TL2 untuk ditangkap dan disimpan ke RCAP2H dan RCAP2L. sebagai tambahan transisi tinggi ke rendah

pada T2EX menyebabkan bit EXF2 pada T2CON diset. Bit EXF2 sama halnya dengan bit TF2 dapat menghasilkan sela.



**Gambar 2-12.** Timer 2 pada Mode Capture

(Sumber : Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

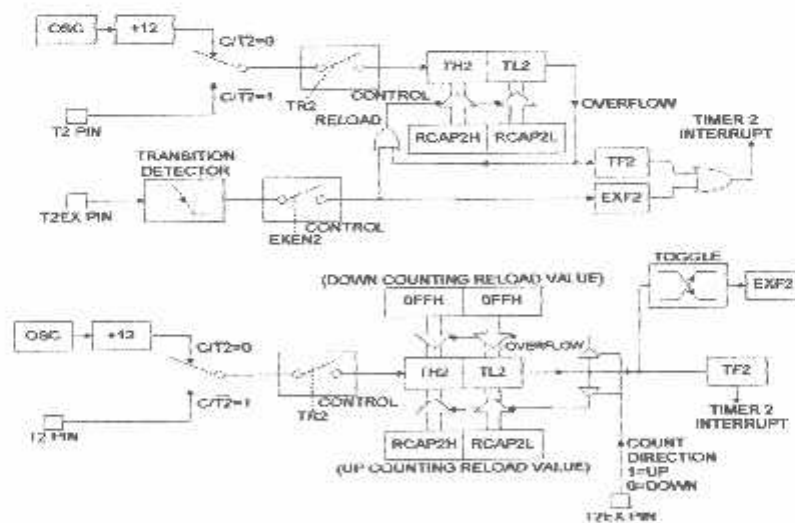
#### 2.3.5.2. Auto Reload (Up/Down Counter)

Timer 2 dapat diprogram untuk menghitung naik (maju) atau menghitung mundur jika dikonfigurasi sebagai mode 16 bit auto reload. Fitur ini dapat dimatikan dengan mengatur bit DCEN (*Down Counter Enable*) pada SFR T2MOD. Pada saat reset DCEN akan berlogika rendah maka Timer 2 akan memiliki default untuk menghitung maju (*Up Counter*). Jika bit DCEN diset, Timer 2 dapat menghitung maju atau mundur tergantung pada nilai logika pada pin T2EX.

Pada gambar berikut Timer 2 secara otomatis menghitung naik (maju) pada saat DCEN=0. Pada mode ini, 2 pilihan dipilih oleh bit EXEN2 pada SFR T2CON. Jika EXEN2=0, Timer 2 akan naik menjadi 0FFFFH dan kemudian akan mengeset bit TF2 jika telah melimpah (*overflow*). *Overflow* juga menyebabkan terjadinya

register timer diisi kembali dengan nilai 16 bit dari RCAP2H dan RCAP2L. Jika EXEN2=1 suatu 16 bit *reload* dapat diaktifkan (di-trigger) baik oleh *overflow* atau oleh transisi tinggi ke rendah pada pin T2EX. Transisi ini juga akan mengeset bit EXF2. Kedua bit TF2 dan EXF2 dapat menimbulkan sela jika diaktifkan.

Mengeset bit DCEN akan mengaktifkan Timer 2 untuk menghitung naik atau mundur. Pada mode ini pin T2EX akan mengontrol arahnya (maju/mundur). Suatu logika '1' pada pin T2EX membuat Timer 2 menghitung maju (*count up*). Timer akan melimpah pada hitungan 0FFFFH dan mengeset bit TF2. Overflow ini juga menyebabkan nilai 16 bit pada RCAP2H dan RCAP2L diisi kembali (*reloaded*) ke register timer yaitu TH2 dan TL2. Suatu logika '0' pada pin T2EX membuat Timer 2 menghitung maju (*count down*). Timer akan mengalami underflow saat TH2 dan TL2 sama dengan nilai yang tersimpan pada RCAP2H dan RCAP2L. underflow ini akan mengeset bit TF2 dan menyebabkan 0FFFFH akan diisi kembali (*reloaded*) ke timer register.



**Gambar 2-13.** Timer 2 pada Mode Auto Reload DCEN=0 (atas), DCEN=1 (bawah)  
(Sumber : Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

### 2.3.5.3. Baud Rate Generator

Timer 2 dapat dipilih sebagai *baud rate generator* dengan mensetting TCLK dan RCLK pada SFR T2CON. *Baud rate* untuk pengiriman dan penerimaan dapat berbeda jika Timer 2 digunakan untuk penerimaan atau pengiriman, sementara Timer 1 digunakan untuk tugas lain.

*Baud rate generator* secara teknis sama dengan mode *auto reload*, dimana pelimpahan TH2 menyebabkan register Timer 2 diisi kembali dengan nilai 16 bit pada register RCPA2H dan RCPA2L yang telah diisi (*preset*) oleh program pemakai.

*Baud rate* pada mode 1 dan 3 ditentukan oleh *rate overflow Timer* berdasarkan persamaan :

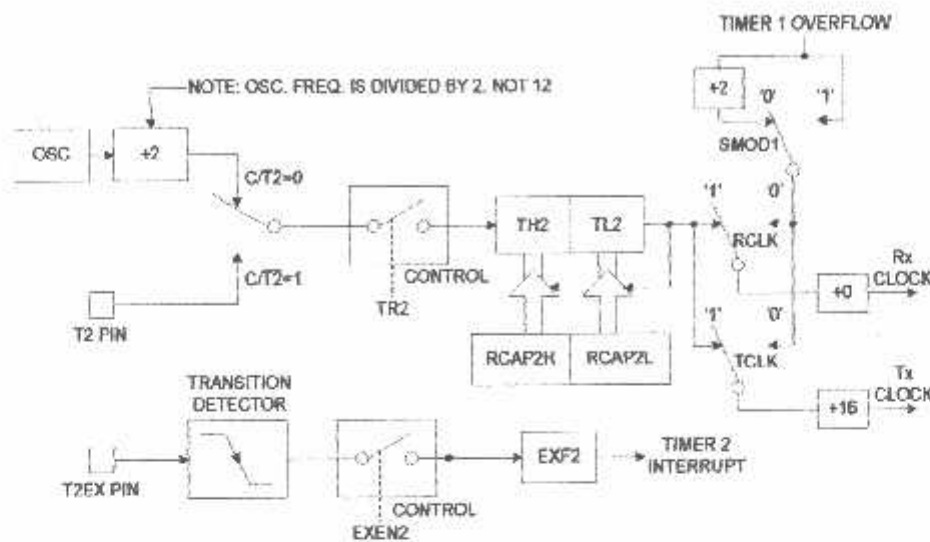
$$\text{Baud Rate [mode 1,3]} = \frac{\text{Timer2OverflowRate}}{16} \dots\dots\dots(2-19)$$

Timer dapat dikonfigurasi sebagai operasi timer atau counter. Pada kebanyakan pemakaian adalah difungsikan sebagai timer dimana bit CP/T2 dibuat '0'. Operasi timer adalah berbeda jika Timer 2 difungsikan sebagai *baud rate generator*. Secara normal sebagai timer, akan naik setiap siklus mesin (pada 1/12 frekuensi osilator). Sebagai *baud rate generator* akan dinaikkan tiap *state time* (pada 1/2 frekuensi osilator). Rumus untuk *baud rate* adalah sebagai berikut :

$$\text{Baud Rate [mode 1,3]} = \frac{\text{FrekuensiOsilator}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \dots\dots\dots(2-20)$$

Dimana (RCAP2H,RCAP2L) adalah isi dari register RCAP2H dan RCAP2L yang diambil sebagai 16 bit integer tak bertanda.





**Gambar 2–14.** Timer 2 sebagai *Baud Rate Generator*

(Sumber :Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

#### 2.3.5.4. Programmable Clock Out

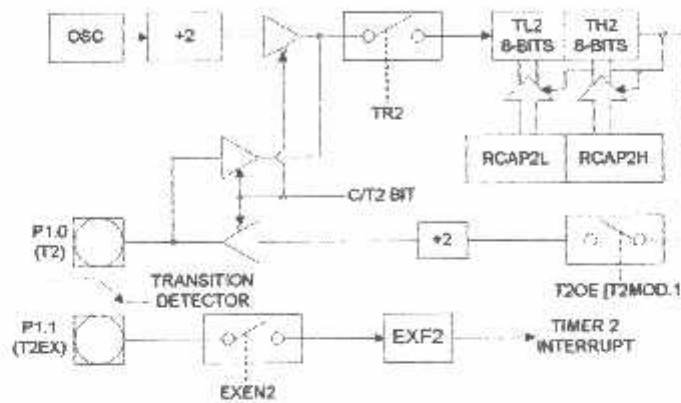
Fungsi terakhir dari Timer 2 adalah untuk menghasilkan suatu detak/pulsa (*clock*). Detak dengan siklus tugas 50% dapat diprogramkan sehingga keluar dari P1.0. Pin ini selain sebagai pin I/O biasa juga memiliki dua fungsi alternative. Dapat diprogram sebagai pin masukan untuk *eksternal clock Timer/Counter 2* atau sebagai penghasil clock dengan siklus tugas 50% dengan rentang dari 61 Hz sampai 4 MHz jika dipakai 16 MHz kristal sebagai frekuensi mikrokontroler.

Untuk membuat *Timer/Counter 2* sebagai generator detak, bit C/T2 (T2CON) harus dibuat rendah dan bit T2OE (T2MOD.1) harus diset. Bit TR2 (T2CON.2) sebagai *start/stop timer*.

Frekuensi detak yang dihasilkan bergantung kepada frekuensi osilator yang dipakai dan nilai reload pada *capture register Timer 2* (RCAP2H,RCAP2L) seperti dalam persamaan berikut ini :

$$\text{Frekuensi Clock Out} = \frac{\text{Frekuensi Osilator}}{4 \times [65536 - (RCAP2H, RCAP2L)]} \dots (2-21)$$

Gambar berikut ini menunjukkan Timer 2 sebagai *clock out generator* :



**Gambar 2-15.** Timer 2 dalam *Clock Out Mode*

(Sumber : Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

### 2.3.6. Sistem Interupsi

Mikrokontroler AT898252 mempunyai 5 buah sumber interupsi yang dapat membangkitkan permintaan yaitu INT0, INT1, T1, T2 dan Port Serial.

Saat terjadinya interupsi mikrokontroler secara otomatis akan menuju ke subrutin pada alamat tersebut. Setelah interupsi selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Tiap-tiap sumber interupsi dapat *enable* atau *disable* secara software.

Tingkat prioritas semua sumber *interrupt* dapat diprogram sendiri-sendiri dengan set atau clear bit pada (*interrupt priority*). Jika dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan akan dilakukan polling untuk menentukan mana yang akan dilayani.

**Tabel 2-5. Alamat Sumber Interupsi**

Sumber Interupsi	Alamat Awal
Interrupt Luar 0 ( INT 0 )	03 <sub>H</sub>
Pewaktu/Pencacah 0 ( T0 )	0B <sub>H</sub>
Interrupt Luar 1 ( INT 1 )	13 <sub>H</sub>
Pewaktu/Pencacah 1 ( T1 )	1B <sub>H</sub>
Port Serial	23 <sub>H</sub>

(Sumber :Data Sheet AT89S8252 [www.atmel.com](http://www.atmel.com))

#### 2.4. Interface Serial

AT89S8252 mempunyai *On Chip serial* port yang dapat digunakan untuk komunikasi data *serial* secara *full duplex* yang berarti bisa mengirim dan menerima secara bersamaan. *Register port serial* menerima dan mengirim diakses melalui SBUF. Port serial dapat digunakan secara sinkron / asinkron. Komunikasi serial secara sinkron yaitu bentuk komunikasi data serial yang memerlukan sinyal pewaktu sebagian sinkronisasi, sedangkan pada komunikasi asinkron tidak memerlukan pewaktu sebagai sinkronisasi.

##### 2.4.1. Komunikasi Serial Antara PC dengan MCU

IC digital, termasuk mikrokontroler, umumnya bekerja pada level tegangan TTL, yang dibuat atas dasar tegangan catu daya +5V. Rangkaian *input TTL* menganggap tegangan kurang dari 0.8 volt sebagai level tegangan “0” dan tegangan lebih dari 2.0 volt dianggap sebagai level tegangan “1” : level tegangan ini sering dikatakan sebagai level tegangan TTL. Sedangkan pada PC / Serial Port tegangan antara +3 sampai +15 volt dianggap sebagai level tegangan “1”. Dari percobaan

acuan tegangan tersebut diperlukan RS-232 sebagai jembatan untuk menghubungkan antara MCU dengan PC, sehingga transfer data dapat dilakukan.

#### 2.4.2. Interface Unit RS-232

Dalam komunikasi data komputer terdapat dua *terminal* ( *port* ) komunikasi yaitu *port paralel* ( *LPT standard* ) dan *port serial* ( *RS-232 standard* ) yang masing-masing memiliki *spesifikasi* yang berbeda. Biasanya *standard serial* RS-232 memiliki dua buah *terminal* yang sering disebut COM 1 dan COM 2.

DTE ( *Data Terminal Equipment* ) atau *terminal data* yang melakukan pertukaran *data seri* dan menggunakan *data biner*, dapat menggunakan *terminal* komunikasi seri RS-232 yang dibuat oleh EIA ( *Electrical Industry Association* ) dan dikenal dengan *standard CCITT V.24*. Sebuah DTE adalah perangkat *prosessor* yang dilengkapi dengan *data paralel* menjadi *serial* dan sebaliknya.

Sedang yang dimaksud dengan DCE ( *Data Communication Equipment* ) adalah perangkat yang mengubah *data serial* menjadi salah satu bentuk sinyal *analog* yang dapat ditransformasikan pada saluran *transmisi* seperti telepon atau radio. *Standard* RS-232 ini berisikan karakteristik sinyal listrik, karakteristik *mekanik* dan cara rangkaian fungsionalnya.

- Tegangan rangkaian terbuka ( *open loop* ) tidak boleh lebih dari 25 Volt.
  - Keadaan logika "1" ( *Mark* ) pada *driver* ditandai dengan tegangan antara -5 sampai -25 Volt.
  - Keadaan logika "0" ( *Space* ) pada *driver* ditandai dengan tegangan antara +5 sampai +25 Volt.
-

- *Slew Rate* ( Perubahan tegangan keluaran perancangan satuan waktu ) < 30 V/s.

Waktu untuk melewati daerah *invalid* -3 V hingga +3 V ≤ 1 ms.

Karakteristik mekanik *interface* ditentukan dengan konektor DB-25 pin atau DB-9 pin, dimana tiap-tiap kontak konektor memiliki fungsi tertentu seperti tabel2-6.

**Tabel 2-6.** Nama Pin dan Keterangan dari *Port* RS-232

Nama Sinyal	Arah Sinyal	Nomor kaki konektor	
		DB9	DB25
Signal Common	-	5	7
Transmitted Data (TD)	Ke DCE	3	2
Received Data (RD)	Dari DCE	2	3
Received Data (RTS)	Ke DCE	7	4
Celar to Send (CTS)	Dari DCE	8	5
DCE Ready (DSR)	Dari DCE	6	6
DTE Ready	Ke DCE	9	22
Ring Indicator (RI)	Dari DCE	9	22
Data Carrier Detect (DCD)	Dari DCE	1	8

(*Sumber: Interfacing Komputer dan Mikrokontroler*)

Fungsi pin DB - 9 RS – 232 adalah :

**1. *Data Carrier Detect* ( DCD )**

Berguna pada DTE untuk tidak memperbolehkan penerimaan data.

**2. *Transmitted Data* ( TXD )**

Berguna untuk jalur pengiriman *data* dari DTE ke DCE.

**3. Received Data ( RXD )**

Berguna untuk jalur pengiriman *data* dari DCE ke DTE.

**4. Data Terminal Ready ( DTR )**

Berguna untuk memberitahu DCE bahwa DTE telah *aktif* dan siap untuk bekerja.

**5. Signal Ground ( DSR )**

Berguna sebagai referensi semua tegangan *interface*.

**6. Data Set Ready ( DSR )**

Bekerja untuk memberitahu DTE bahwa DCE telah *aktif* dan siap untuk bekerja.

**7. Request to Send ( RTS )**

Bertugas memberitahu DCE bahwa DTE akan mengirimkan data. RTS merupakan sebuah *protokol hardware* yang mendahului pengiriman data DTE ke DCE.

**8. Clear to Send ( CTS )**

Bertugas memberitahu DTE bahwa DCE siap untuk menerima data. CTS merupakan sebuah *protokol hardware* yang mendahului pengiriman data DTE ke DCE.

**9. No Connection ( NC )**

Ditinjau dari proses jembatan ( *handshaking* ) *standard* RS-232 yaitu pin-pin DTR, DSR, RTS, CTS, dan CD maka perangkat lunak akan mengaktifkan DTR yang menandakan bahwa komputer siap melakukan *transmisi* atau menerima data dan apabila *terminal* lain juga dihidupkan

serta siap menerima/ mengirim data, maka *terminal* tersebut akan mengirimkan sinyal DSR.

#### 2.4.2.1. Max 232 Sebagai Pengubah Tegangan TTL

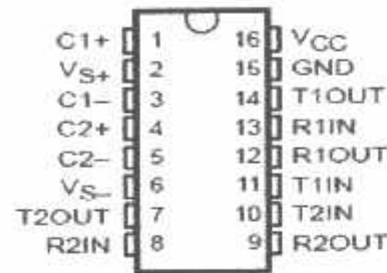
Max 232 merupakan pengubah TTL ke level tegangan RS-232. Max 232 memiliki dua *driver* mengonversikan RS-232 ke level TTL, dan dua penerima yang merubah TTL ke RS-232. Max 232 memiliki 16 pin dan dioperasikan dengan empat buah kapasitor yang memiliki nilai 1uF.

RS-232 merupakan salah satu jenis antar muka (*interface*) dalam proses transfer data antar komputer dalam bentuk serial transfer. RS-232 merupakan singkatan dari *Recommheded Standard number 232*. Alat ini dibuat oleh *Electronic Industry Assosiation*, untuk *Interface* antara peralatan *terminal data* dan peralatan komunikasi data, dengan menggunakan *data binmer serial* sebagai data yang ditransmisikan IC MAX 232 ini mempunyai empat buah bagian konverter yaitu dua buah *driver receiver* dan dua buah *driver transmitter*.

Saluran data pada port seri PC menggunakan *standard* RS-232, dimana *logic 0 (low)* dinyatakan sebagai tegangan antara +3 volt sampai +10 volt, dan *logic 1 (high)* dinyatakan sebagai tegangan antara -3 volt sampai -10 volt. Level tegangan ini tidak sesuai dengan tegangan *level* tegangan yang dipakai pada port seri 51 AT atau port seri AT8252 yang menggunakan *Standard TTL (Transistor Transistor Logic)*, yaitu *level* tegangan baku dalam rangkaian – rangkaian *digital*.

Dalam *standard* TTL, *logic 0 (low)*, dinyatakan sebagai tegangan antara 0 volt sampai 0.8 volt, dan *logic 1 (high)* dinyatakan sebagai tegangan antara 3.5 volt sampai 5 volt. Untuk dua MCU yang dihubungkan secara *serial* pada jarak tertentu

maka dibutuhkan IC MAX 232 karena *level* tegangan TTL terlalu kecil untuk ditransfer.



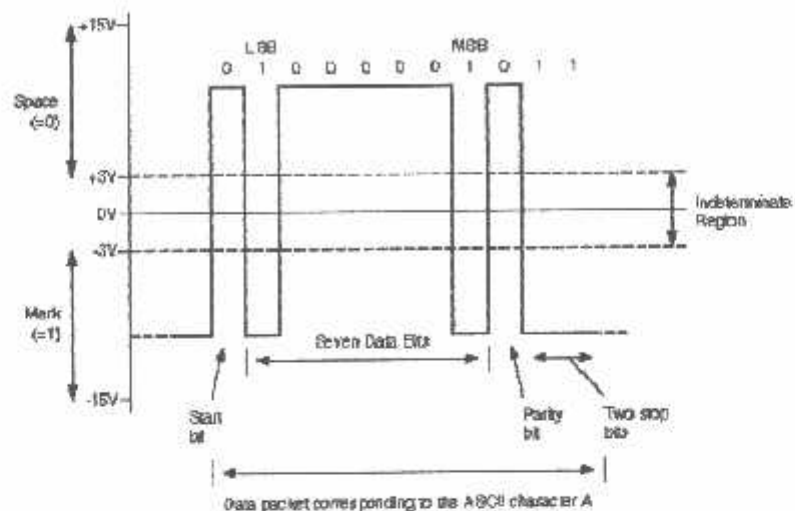
**Gambar 2-16.** IC MAX 232

(Sumber : [www.electroniclab.com](http://www.electroniclab.com) DataSheet MAX 232)

RS-232 merupakan seperangkat alat yang berfungsi sebagai *interface* dalam proses *transfer* data secara *serial*. Metode pengiriman secara *serial* RS-232 adalah *asinkron*. Pengiriman *asinkron* berarti tidak membutuhkan pewaktu sebagai *sinkronisasi*. Dalam pengiriman secara *serial asinkron*, clock dapat dikirimkan, tetapi dikondisikan oleh *timing start bit* yang merupakan isyarat dari sumber ke tujuan untuk mendekodekan adanya pengiriman karakter sudah selesai dikirim.

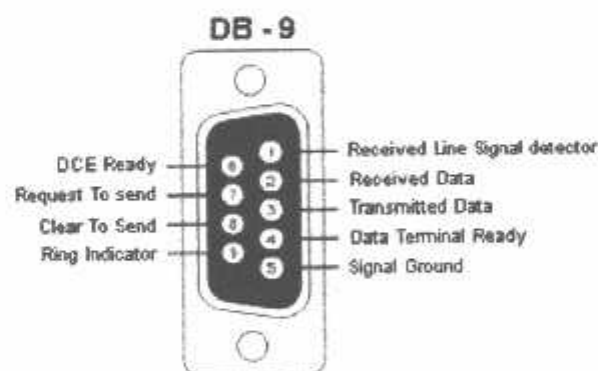
Karakteristik *electris* dan sistem RS-232 adalah mempunyai tegangan keluaran antara -15 volt sampai dengan +15 volt. Tegangan +3 sampai +15 volt untuk logika '0' / *spacing* dan tegangan -3 sampai -15 volt untuk logika '1' / *marking*. Hal tersebut dinyatakan dalam gambar 2-17.





**Gambar 2-17. 1 byte of Async Data**  
(Sumber : [www.arcelect.com/rs232.htm](http://www.arcelect.com/rs232.htm))

Didalam komputer terdapat fasilitas komunikasi *serial* yang menggunakan *Standard RS-232*, yaitu terletak pada COM 1 dan COM 2. Kedua fasilitas ini menggunakan konektor DB-9 atau DB-25. Gambar konektor DB-9 seperti terdapat dalam gambar 2-15.



**Gambar 2-18. Konektor DB-9**  
(Sumber : *Interfacing Komputer dan mikrokontroler*)

Fungsi masing – masing pin pada DB-9 seperti terdapat dalam tabel 2-5

**Tabel 2-7.** Fungsi Pin RS-232 dalam DB-9

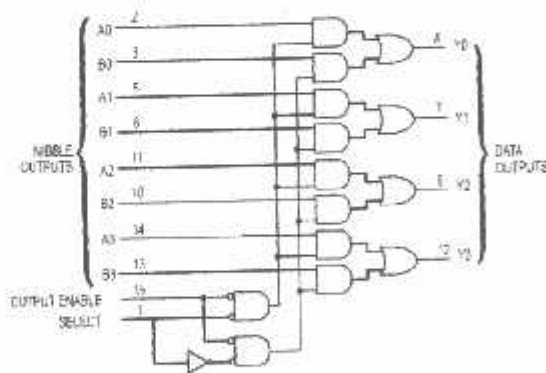
Pin	Nama	Fungsi
1	DCD (Data Carrier Detect)	Mendeteksi Sinyal Carrier dari modem lain
2	RD (Received Data Line) / (RXD)	Pengiriman data serial dari DCE ke DTE
3	TD (Transmitted Data Line) / (TXD)	Pengiriman data serial dari DTE ke DCE
4	DTR (Data Terminal Ready)	Memberitahukan DCE bahwa DTE telah aktif dan siap untuk bekerja
5	Ground	Referensi semua tegangan antar muka
6	DSR (Data Set Ready)	Memberitahukan DTE bahwa DCE telah aktif dan siap untuk bekerja
7	RTS (Request To Send)	Memberitahukan DCE bahwa DTE akan mengirim data
8	CTS (Clear To Send)	Memberitahukan DTE bahwa DCE siap menerima data
9	RI (Ring Indikator)	Aktif jika <i>modem</i> menerima sinyal ring jalur telepon

(Sumber : *Interfacing Komputer dan mikrokontroler*)

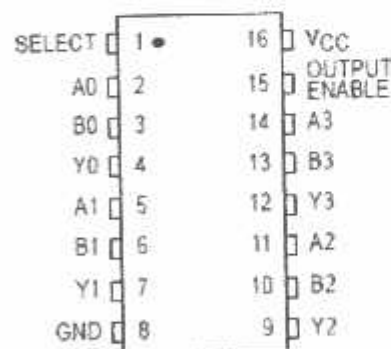
Jalur data (TXD dan RXD) untuk transport data, TXD adalah jalur *output* komputer, data dikirim dari pin ini. Sedangkan RXD adalah penerima untuk komputer, data yang datang akan diterima oleh pin ini. Pin ke empat adalah *output* (RTS) dimana sebuah sinyal akan diberikan pada alat yang dihubungkan dengan maksud meminta kiriman data. CTS adalah sinyal masukan yang menunggu sinyal dari alat yang terhubung ketika alat tersebut menerima sinyal RTS dan bisa menerima data maka ia akan mengirimkan sinyal balik yang merupakan CTS. DTR adalah sinyal keluaran yang memberi tanda bahwa ada alat yang terhubung dan akan mengirimkan data. DSR merupakan sinyal input yang mana jika alat yang terhubung menerima sinyal DTR ia akan memberi sinyal balik kemudian diterima sebagai sinyal DSR.

## 2.5. IC 74HC157A

IC 74HC157A adalah IC dengan Quad 2-input Multiplexer. Input data 4 bit dari 2 sumber dengan memberikan kondisi high/ low pada common Select Input dan Output Enable akan menghasilkan output sesuai yang dibutuhkan.



**Gambar 2-19.** Logic Diagram



**Gambar 2-20.** IC 74HC157

(Sumber : Data Sheet 74HC157A)

IC 74HC157 bisa dikatakan sebagai sebuah IC selector yang hasil outputnya ditentukan oleh logika yang diberikan pada *common select input*. Maka persamaan logika tersebut dapat dilihat dengan tabel kebenaran dibawah ini :

**Tabel 2-8.** Kebenaran IC 74HC157

OUTPUT ENABLE	SELECT INPUT	INPUTS		OUTPUTS ( Y )
		A	B	
H	X	X	X	L
L	H	X	L	L
L	H	X	H	H
L	L	L	X	L
L	L	H	X	H

X : Don't Care

H : High Voltage Level

L : Low Voltage Level

(Sumber : Data Sheet 74HC157A)

Sehingga dalam perencanaan ini IC tersebut digunakan sebagai selektor mikrokontroler tepatnya pin rx untuk digunakan secara bergantian antara RFID reader dengan RS232. IC 74HC157 dapat terpicu dengan tegangan 5 Volt dan arus keluaran 25mA dengan suhu kerja dari -55°C sampai 125°C.

## 2.6. LCD ( Liquid Crystal Display ) M1632

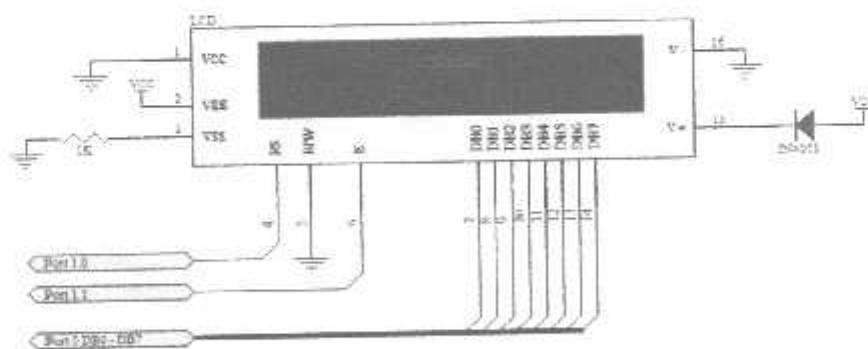
*Liquid Crystal Display* adalah modul tampilan yang mempunyai konsumsi daya yang relatif rendah dan terdapat sebuah kontroller CMOS didalamnya. Kontroller tersebut sebagai pembangkit ROM/RAM dan display data RAM. Semua

fungsi tampilan di kontrol oleh suatu instruksi modul LCD dapat dengan mudah diinterfacekan dengan MPU.

Spesifikasi dari LCD M1632:

- ❖ Terdiri dari 32 karakter yang dibagi menjadi 2 baris dengan display dot matrik 5 X 7 ditambah cursor
- ❖ Karakter generator ROM dengan 192 karakter
- ❖ Karakter generator RAM dengan 8 tipe karakter
- ❖ 80 X 8 bit display data RAM
- ❖ Dapat diinterfacekan dengan MPU 8 atau 4 bit
- ❖ Dilengkapi fungsi tambahan : Display clear, cursor home, display ON/OFF, cursor ON/ OFF, display character blink, cursor shift dan display shift
- ❖ Internal data
- ❖ Internal otomatis dan reset pada power ON
- ❖ 15 V power supply tunggal

(Sumber: LCD Module User Manual)



**Gambar 2-21.** Pin Out LCD M1632  
( Sumber: Perencanaan dan Pembuatan )

LCD modul M1632 mempunyai 16 pin dengan fungsi sebagai berikut :

**Tabel 2-9. Fungsi Pin – Pin LCD**

Nama Pin	Jumlah	I/O	Tujuan	Fungsi
DB0-DB3	4	I/O	MPU	Tri state bidirectional lower data bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus
DB4-DB7	4	I/O	MPU	Tri state bidirectional upper data bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus
E	1	Input	MPU	Sinyal operasi dimulai: sinyal aktif baca/tulis
R/W	1	Input	MPU	Sinyal pilih data dan tulis (0:tulis,1:baca)
RS	1	-	Power supply	Sinyal pilih register 0: Instruction register (write) Busy flag dan address counter (read) 1:Data register (write dan read)
VLC	1	-	Power supply	Penyetelan kontras pada tampilan LCD
VDD	1	-	Power supply	+ 5V
VSS	1	-	Power supply	Ground 0V

(Sumber : LCD Module User Manual)

### 2.6.1. Register

Control LCD mempunyai 2 register 8 bit yaitu *Instruction register* (IR) dan *Data Register* (DR). Kedua register tersebut dipilih melalui *Register Select* (RS). IR menyimpan kode instruksi seperti Display clear dan cursor shift, dan alamat

informasi dari Display Data RAM (DD RAM) dan character generator RAM (CG RAM)

DR menyimpan data sementara untuk ditulis ke DD RAM atau CG RAM, atau dibaca dari DD RAM atau CG RAM. Ketika data ditulis ke DD RAM atau CG RAM dari MPU, data di DR secara otomatis ditulis ke DD RAM atau CG RAM dengan operasi internal. Tetapi ketika data dibaca dari DD RAM atau CG RAM maka alamat data ditulis pada IR. Data tersebut akan dimasukkan ke DR dan MPU akan membaca data dari DR. Setelah operasi pembacaan, alamat berikutnya diset data dari DD RAM atau CG RAM pada alamat tersebut akan dimasukkan ke DR untuk operasi berikutnya.

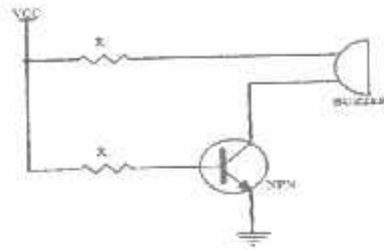
Display data RAM (DD RAM) mempunyai kapasitas area 80 X 8 bit. Beberapa area dari DDRAM yang tidak digunakan untuk display dapat digunakan sebagai General data RAM.

Pada LCD masing-masing pin mempunyai range alamat tersendiri, alamat itu diekspresikan dengan bilangan heksa. Untuk line 1 range alamat berkisar antara 00h-0Fh sedangkan untuk line 2 alamat berkisar antara 40h-4fh

(Sumber : LCD Module User Manual)

## 2.7. BUZZER

Perangkat *Buzzer* digunakan untuk menghasilkan bunyi, merupakan komponen resonator Riezoelectric yang digunakan untuk mengadakan isyarat terdengar sebagai indikator. *Buzzer* akan aktif dengan cara mengeluarkan sinyal suara (berbunyi) dengan lama waktu sesuai dengan perencanaan nanti.



**Gambar 2-22.** Rangkaian *Driver Buzzer*  
( *Sumber: Perencanaan dan Pembuatan* )

## 2.8. Telepon Seluler

Telepon seluler ( *ponsel* ) adalah suatu jenis telepon bergerak yang menggunakan teknologi seluler sebagai akses komunikasinya. Sehingga memudahkan seseorang untuk berkomunikasi dimanapun dan dalam kondisi apapun. Dengan adanya ponsel komunikasi lebih mudah dan lebih efisien. Seiring dengan perkembangan teknologi dibidang *mobile* sistem, ponsel tidak hanya mengirim dan menerima data suara tetapi juga data karakter yang biasa disebut SMS (*Short Message Service*). Masing-masing ponsel memiliki fitur-fitur yang berbeda. Salah satunya adalah bagian *connectors*, ponsel bisa dihubungkan dengan perangkat komputer dengan menggunakan kabel data, *Bluetooth*, *infrared*.

### 2.8.1. Nokia 8250

Nokia 8250 merupakan salah satu merek ponsel GSM yang dapat dihubungkan dengan perangkat komputer. Dengan memanfaatkan *connectors* yang dimiliki ponsel tersebut maka bisa dihubungkan dengan perangkat komputer. Sehingga semua kegiatan yang dilakukan oleh ponsel tersebut, baik mengirim atau menerima suatu pesan singkat ( SMS ) bisa dilakukan pada perangkat komputer.



## 2.9. Borland Delphi

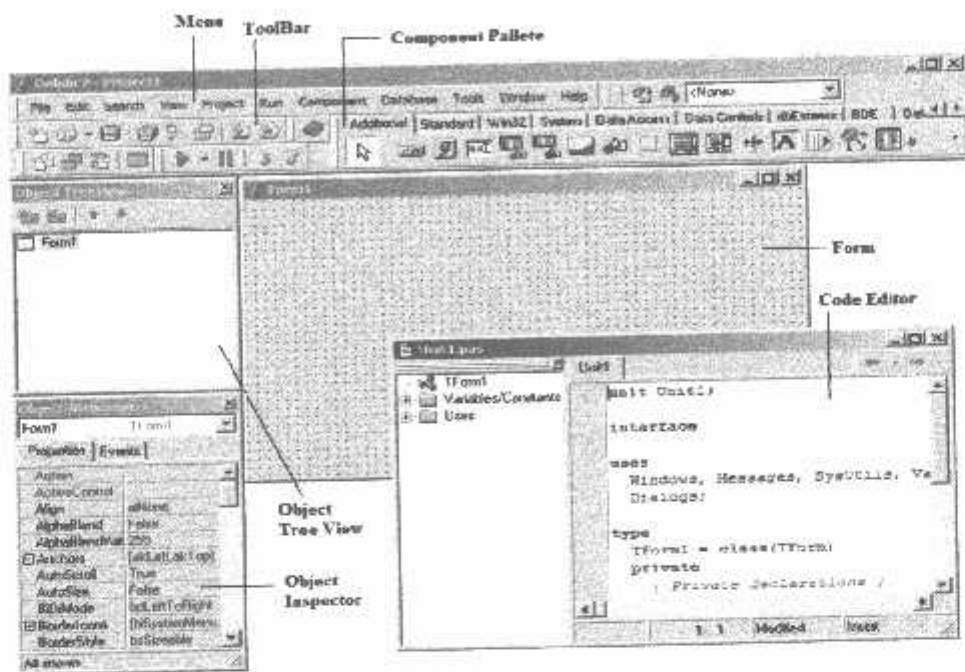
Secara umum, Borland Delphi adalah sebuah program untuk membuat aplikasi – aplikasi berbasis Windows. Bahasa pengembangan yang digunakan oleh Delphi adalah bahasa Pascal. Turbo Pascal dikenal dengan kelebihan dalam kecepatan eksekusi dan kompilasi, dibandingkan dengan bahasa pemrograman lain yang berkembang saat ini. *Integrated Development Enviroment* (IDE) yang diperkenalkan dan diterapkan oleh Turbo Pascal sangat memudahkan para programmer merealisasikan program aplikasi mereka. Dengan IDE seorang programmer dapat dengan cepat dan mudah menulis kode program, melakukan kompilasi, melihat kesalahan (*error*) program, serta langsung menuju letak kesalahan dan memperbaiki kesalahan tersebut. Kemudian Turbo Pascal dirubah menjadi yang berorientasi obyek (*Object Oriented Programming*) berbasis tampilan visual yang menarik, dan dilengkapi kemampuan akses ke basis data. Inilah yang kemudian dikenal sebagai Delphi.

Delphi dapat digolongkan ke dalam bahasa tingkat tinggi (*High Type Language*) karena segala kemudahan ditawarkan untuk perancangan sebuah aplikasi.

( *Sumber: Teknik Antarmuka Mikrokontroler Dengan Komputer Berbasis Delphi, 2006* )

### 2.9.1. IDE (*Integrated Development Enviroment*)

IDE adalah sebuah lingkungan yang berisi tool – tool yang diperlukan untuk desain, menjalankan dan mengetes sebuah aplikasi, disajikan dan terhubung dengan baik sehingga memudahkan pengembangan program. Di Delphi, IDE terdiri dari :



**Gambar 2-23. IDE (Integrated Development Enviroment)**  
(Sumber: Visual Borland Delphi 7, 2004)

#### a. Main Window

Main Window adalah bagian utama dari IDE. Main Window mempunyai semua fungsi utama dari program – program Windows lainnya.

#### b. Menu Utama

Menu utama dipakai untuk membuka atau menyimpan file, memanggil wizard, menampilkan jendela lain, mengubah option dan lain sebagainya.

#### c. Toolbar

Dengan menu toolbar dapat melakukan beberapa operasi pada menu utama yang setiap tombol berisi informasi mengenai fungsi dari tombol tersebut.

#### d. Form Designer

Jendela kosong yang digunakan untuk merancang aplikasi Windows.

**e. Code Editor**

Merupakan bagian yang terpenting di lingkungan Delphi. Jendela ini dipakai untuk menuliskan program Delphi.

**f. Code Explorer**

Code explorer digunakan untuk memudahkan navigasi didalam file unit.

**g. Object Treeview**

Merupakan daftar dari komponen-komponen apa saja yang telah kita pergunakan dan juga merupakan peta dari program yang kita buat.

(Sumber: Visual Borland Delphi 7, 2004)

**2.9.2. Menu Borland Delphi****1. Menu File**

Berisi fasilitas untuk membuat Project baru, menyimpan Project, membuka Project, dan keluar dari IDE Delphi.

**2. Menu Edit**

Berisi fasilitas untuk melakukan *editing* atau perubahan pada kode program, juga pengaturan form dan unit ( ukuran, penempatan, kontrol, dsb ).

**3. Menu Search**

Berisi Fasilitas untuk melakukan pencarian atau penggantian kata dalam tubuh kode program ( unit ) dan juga mencari letak kesalahan program.

**4. Menu View**

Berisi fasilitas untuk mengatur tampilan IDE Delphi. Misalnya Object Inspector, daftar komponen, pengaturan *Toolbar*, Form, dan Unit.

#### 5. *Menu Project*

Berisi fasilitas yang berkaitan dengan properti dari Project, misalnya menambahkan atau memisahkan Form dan Unit dari sebuah Project.

#### 6. *Menu Run*

Berisi fasilitas untuk Kompiler Delphi, yang terpenting adalah *Run* dan *Reset*.

#### 7. *Menu Component*

Berisi fasilitas untuk mengatur properti *Component Pallete* dan instalasi komponen baru.

#### 8. *Menu Database*

Berisi fasilitas yang berkaitan dengan pembuatan aplikasi data.

#### 9. *Menu Tools*

Berisi fasilitas untuk melakukan pengaturan direktori, *library*, *path* penyimpanan file-file penting dalam Delphi, dan tools yang bekerjasama dengan Delphi.

#### 10. *Menu Window*

Berisi fasilitas untuk berpindah dari satu jendela kerja ke jendela kerja yang lain dalam IDE Delphi.

#### 11. *Menu Help*

Berisi fasilitas menerima bantuan atau keterangan tentang Delphi.

( *Sumber: Teknik Antarmuka Mikrokontroler Dengan Komputer Berbasis Delphi. 2006* )

### 2.9.3. **Komponen Visual dan Nonvisual**

*Visual Component Library* (VCL) adalah herarki dari *class* ( *object* ) yang ditulis dengan *object Pascal* dan terdapat dalam IDE Delphi. Dengan *Component*

*Palette* dan *Object Inspector*, kita dapat menempatkan komponen *VCL* pada form dan memanipulasi propertinya tanpa menuliskan program.

*VCL* adalah *Library Delphi* dalam lingkungan Windows. Delphi membagi komponen-komponen ke dalam dua kelompok besar berdasarkan visualisasinya, yaitu:

### 1. Komponen Visual

Yaitu komponen yang akan ditampilkan di layar pada saat aplikasi dijalankan. Komponen visual dipakai untuk membangun antarmuka dengan pemakai. Komponen ini dapat diubah ukurannya dengan cara drag pada salah satu kotak kecil yang ada di sekeliling komponen. Contoh komponen visual adalah Button, Edit, Memo, dan lain-lain.

### 2. Komponen Nonvisual

Yaitu komponen yang tidak ditampilkan di layar pada saat aplikasi dijalankan, tetapi diperlukan dalam aplikasi. Ukuran komponen ini tidak dapat diubah.

( Sumber: Visual Borland Delphi 7, 2004 )

#### 2.9.4. Komponen *TOxygenSMS*

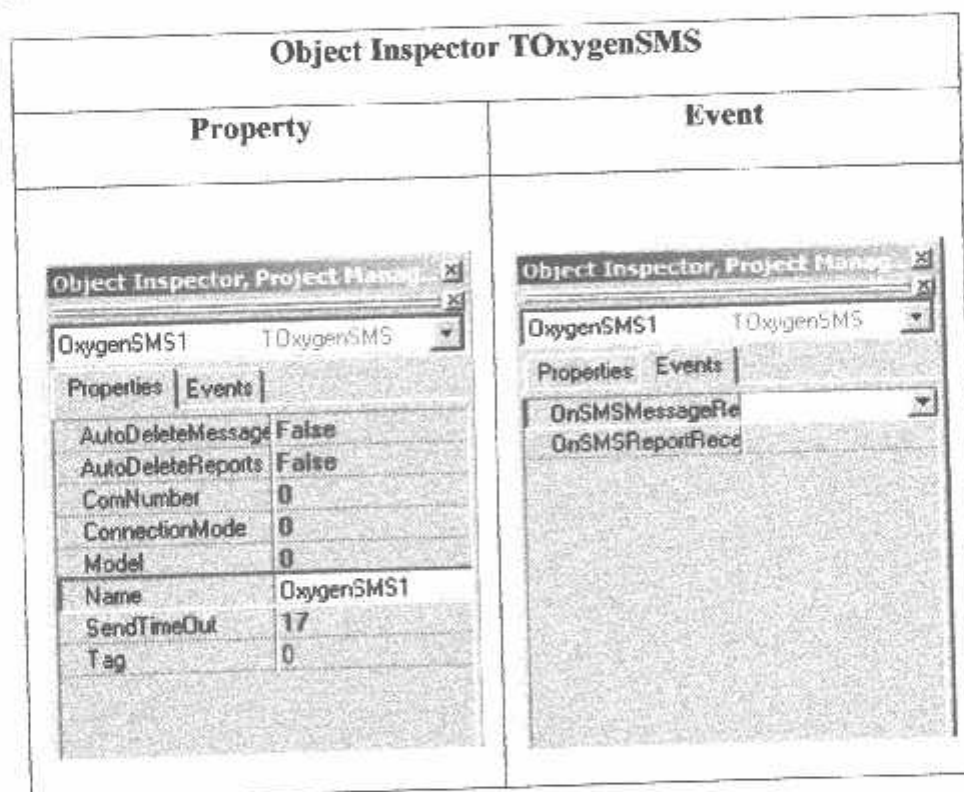
Komponen yang menghubungkan perangkat komputer dan perangkat GSM Nokia. Komponen ini sangat mudah diprogram untuk manajemen SMS, namun hanya mendukung *handset* Nokia saja. Tipe – tipe *handset* Nokia yang didukung adalah :

3210, 3310, 3330, 3390, 3410, 3510, 5110, 5130, 5190, 5210, 6110, 6130, 6150, 6190, 6210, 6250, 6310, 6310i, 6360, 6510, 7110, 7190, 8210, 8290, 8250, 8310, 8390, 8810, 8850, 8855, 8890, 8910;

(Sumber : Aplikasi SMS Untuk Berbagai Keperluan)

#### 2.9.4.1. Object Inspector TOxygenSMS

Pada *Object Inspector TOxygenSMS* memiliki *Property* dan *Event* yang dapat diset sesuai kebutuhan.



**Gambar 2–24.** *Object Inspector TOxygenSMS*  
(Sumber : Perancangan dan Pembuatan)

Secara sederhana properti-properti utama yang harus diatur adalah sebagai berikut :

1. Set Utama:

❖ **ConnectionMode:** Untuk menentukan jenis koneksi yang akan digunakan. Beberapa model konektor seperti sebagai berikut:

- 0 – DAU-9P
- 1 – DLR-3
- 2 – Infrared

Bila menggunakan kabel generik F-Bus/M-Bus maka menggunakan 0, koneksi menggunakan DAU-9P.

❖ **ComNumber:** Isi dengan nomor *Port Com* ( *Serial Port* ) yang digunakan kabel. Bila menggunakan koneksi *Infrared* ( *IrDA* ), properti ini tidak perlu diset.

2. Set Tambahan:

❖ **AutoDeleteMessage**

Properti ini bila di set True akan otomatis menghapus SMS yang telah dibaca.

❖ **AutoDelete**

Tergantung pada *setting* HP yang digunakan, bila HP di set untuk selalu memberikan laporan status SMS ( *Send* / *Pending* / *Receive* ) maka Properti ini di Set True maka setiap *Report* akan otomatis dihapus.

( Sumber : Aplikasi SMS Untuk Berbagai Keperluan )

### 2.9.4.2. Fungsi Internal *TOxygenSMS*

#### ❖ *ReadSMSMessage*

*ReadSMSMessage* secara sederhana adalah fungsi untuk membaca sebuah pesan SMS tertentu dari daftar.

```
ReadSMSMessage ( Index: Longint; var IsMsg: WordBool;  
var Time: TdateTime; var Tex, Send: String; var Delivery:  
integer; var Picture: Tbitmap ) Boolean;
```

Penjelasan parameter :

<b>Index</b>	Posisi pesan SMS yang akan dibaca, parameter ini bergantung pada jumlah <i>memori slot</i> yang disediakan oleh <i>provider</i> , seperti Telkomsel-Simpati menyediakan 24 <i>slot</i> untuk SMS yang dapat disimpan dalam <i>SIM Card</i> .
<b>IsMg</b>	Parameter ini menunjukkan apakah isi <i>slot</i> tersebut adalah sebuah pesan SMS (bernilai TRUE) atau hanya sebuah SMS (bernilai False).
<b>Time</b>	Isi keterangan tanggal dan waktu penerimaan
<b>Text</b>	Bila isi <i>slot</i> adalah pesan SMS, maka pesan tersebut dapat dibaca pada parameter ini.
<b>Send</b>	Nomor Pengirim
<b>Delivery</b>	Bila isi <i>slot</i> adalah <i>Report SMS</i> , maka parameter ini dapat bernilai salah satu dari: 0 : delivered      1 : not deleivered 2 : in process      3 : not deleivered
<b>Picture</b>	Bila pesan SMS berisi gambar, maka gambar tersebut dapat diakses dari parameter ini.



❖ *SendSMSMessage*

Fungsi *SendSMSMessage* secara sederhana adalah fungsi untuk Mengirim SMS.

```
SendSMSMessage ( PhoneNumber: string; Text: string;
Validity: byte; NeedReport, IsUnicode: Boolean; Picture:
Tbitmap ): Boolean;
```

## Penjelasan parameter :

PhoneNumber	Isikan dengan Nomor Handset yang akan dikirim NOTE: Gunakan selalu format kode negara internasional, format internasional indonesia adalah “ +62 “ , jadi seperti nomor tujuan adalah 081334710410 maka ubahlah 0 ( nol ) disebelah depan nomor tersebut menjadi +6281334710410.														
Text	Isi parameter ini dengan pesan SMS yang akan dikirim.														
Validity	Nilai tenggang waktu SMS akan disimpan di <i>server provider</i> ( sampai pesan tersebut terkirim) Nilai ini berkisar kurang lebih: <table><tr><td>1..43</td><td>5menit – 12jam</td><td>Kelipatan 5 menit</td></tr><tr><td>144..167</td><td>12jam – 24jam</td><td>Kelipatan 1 jam</td></tr><tr><td>168..196</td><td>2 hari – 30 hari</td><td>Kelipatan 1 hari</td></tr><tr><td>197 - 255</td><td>5minggu – 63minggu</td><td>Kelipatan 1 minggu</td></tr></table> Umumnya <i>Provider</i> Indonesia menetapkan 167 ( 24 jam ).			1..43	5menit – 12jam	Kelipatan 5 menit	144..167	12jam – 24jam	Kelipatan 1 jam	168..196	2 hari – 30 hari	Kelipatan 1 hari	197 - 255	5minggu – 63minggu	Kelipatan 1 minggu
1..43	5menit – 12jam	Kelipatan 5 menit													
144..167	12jam – 24jam	Kelipatan 1 jam													
168..196	2 hari – 30 hari	Kelipatan 1 hari													
197 - 255	5minggu – 63minggu	Kelipatan 1 minggu													
NeddReport	Isikan <b>True</b> bila ingin mendapatkan status <i>Report</i> SMS, dan <b>False</b> bila tidak perlu.														
IsUnicode	Isikan <b>True</b> bila hendak mengirim menggunakan														

	format SMS Unicode, untuk Indonesia isikan <b>False</b> untuk default.
<b>Picture</b>	Bila Pesan SMS berisi gambar, maka gambar tersebut dapat di set pada parameter ini, isikan <b>NIL</b> bila tidak mengirim.

*(Sumber : Aplikasi SMS Untuk Berbagai Keperluan)*

### BAB III

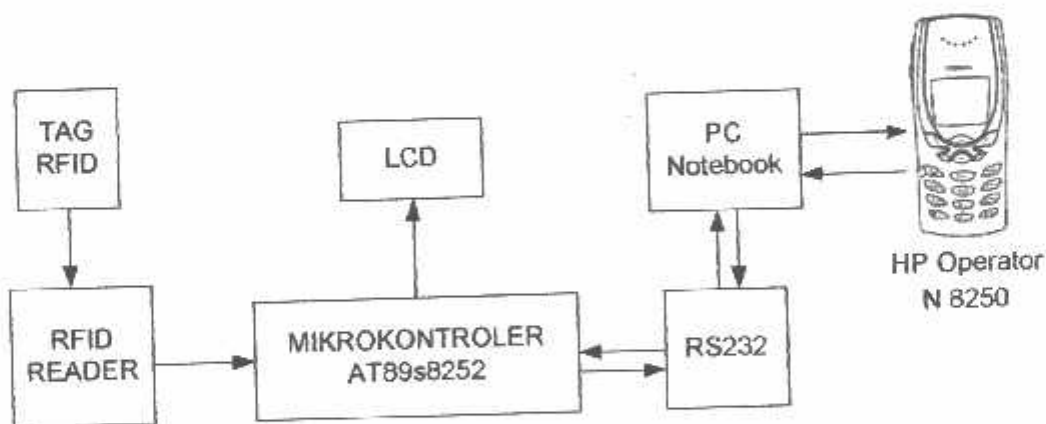
## PERANCANGAN DAN PEMBUATAN ALAT

### 3.1. Pendahuluan

Dalam bab ini akan membahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan dalam sistem absensi pada murid sekolah menggunakan *RFID* yang diinformasikan via sms (*short message service*).

#### 3.1.1. Blok Diagram Keseluruhan Sistem

Perancangan dan pembuatan alat ditunjukkan dengan gambar blok diagram dibawah ini :



**Gambar 3-1.** Diagram Blok Keseluruhan Sistem  
(Sumber : Perancangan dan Pembuatan)

Keterangan fungsi dari masing-masing blok diagram diatas sebagai berikut :

- **Tag RFID**

*Devais* yang menyimpan informasi untuk identifikasi objek.

- **RFID Reader**

*Devais* yang kompatibel dengan *tag* RFID yang akan berkomunikasi secara *wireless* dengan *tag*. Digunakan tipe ID-12 sebagai RFID reader pada perancangan ini.

- **LCD**

Digunakan sebagai penampil data yang dibcrikan oleh mikrokontroler.

- **Mikrokontroler AT89S8252**

Berfungsi sebagai pengontrol masukan dan keluaran data.

- **PC Notebook ( Laptop )**

Digunakan sebagai penampil data beberapa alat yang sedang bekerja, yang ditampilkan pada layar monitor melalui *serial* RS 232 untuk berkomunikasi dengan mikrokontroler dan melalui *infrared* untuk berkomunikasi dengan *Ponsel*.

- **RS 232**

Berfungsi sebagai *interface* antara mikrokontroler dengan *PC Notebook*.

- **Ponsel N 8250**

Sebagai HP operator yang mengirimkan informasi melalui sms(*short message service*).

### 3.2. Prinsip Kerja Alat

Setiap siswa sekolah memiliki *Tag* RFID sebagai kartu absensi dengan kode yang berbeda-beda. *Tag* RFID dapat memancarkan gelombang radio dan diterima oleh *RFID reader*. Gelombang radio tersebut membawa kode- kode yang akan diproses oleh mikrokontroler dan dikirim pada *PC* melalui komunikasi *serial* RS-232.

Selanjutnya data diterima oleh *PC*, diproses dan dicocokkan dengan data yang tersimpan pada *PC* dan data tersebut juga akan ditampilkan pada *LCD*. Apabila data telah sama maka melalui media transmisi baik menggunakan (*Infrared*, *Bluetooth*, atau kabel data) *PC* akan memerintahkan *handphone* operator untuk mengirimkan SMS (*Short Message Service*) ke *handphone* penerima yaitu orang tua/ wali murid masing- masing siswa.

### 3.3. Perancangan Perangkat Keras (*Hardware*)

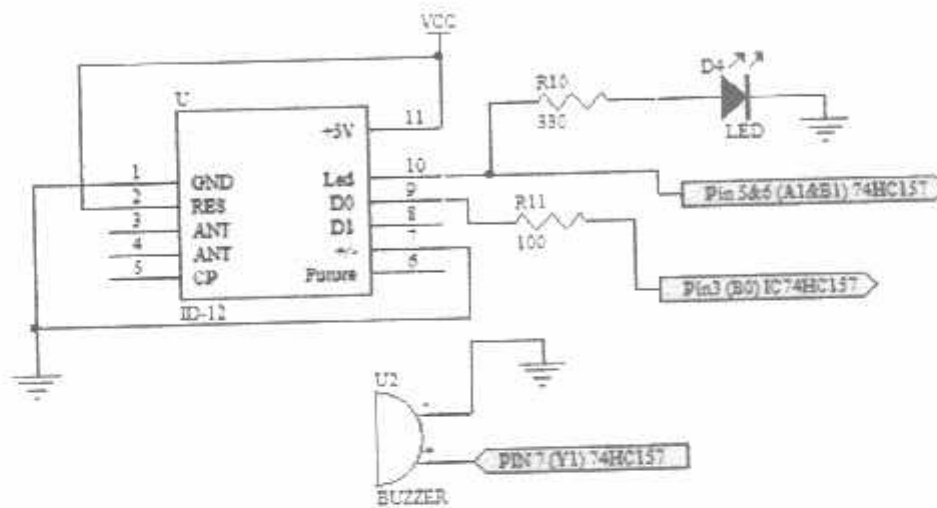
#### 3.3.1. RFID (*Radio Frequency Identification*)

Pada perencanaan sistem alat ini menggunakan *tag pasif*, yaitu *tag* yang tidak memiliki catu daya sendiri serta tidak dapat menginisialisasi komunikasi dengan *reader*. Ukuran *tag pasif* sekitar 0.4 mm x 0.4 mm. Daya jangkau *RFID tag pasif* agar dapat terbaca oleh *RFID transceiver* adalah mulai dari sekitar 1 cm sampai 6 meter. *Tag* yang digunakan adalah ISO Card GK-4001 dan *RFID reader* nya menggunakan ID-12 dengan frekuensi rendah 125 kHz, format data *output* menggunakan ASCII yang dihubungkan ke mikrokontroler menggunakan komunikasi serial UART. *Tag* ini yang digunakan sebagai kartu absensi yang pada saat kartu ini didekatkan ke pembaca *RFID* maka akan dikenali dan akan

---

teridentifikasi nomor seri yang ada didalamnya, sehingga setiap siswa mempunyai kartu dengan nomor seri yang berbeda. Bentuk asli *tag* masih polos maka untuk model kartu bisa dilapisi dengan berbagai macam variasi.

Rangkaian untuk RFID yang direncanakan sebagai berikut :

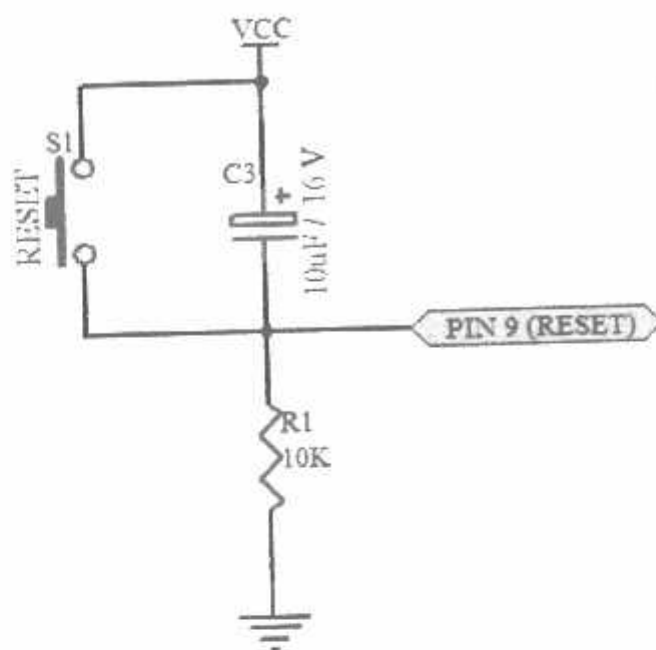


**Gambar 3-2.** Rangkaian RFID Reader  
(Sumber : Perancangan dan Pembuatan)

### 3.3.2. Mikrokontroler AT89S8252

#### 3.3.2.1. Perancangan Rangkaian Reset

Untuk *mereset* mikrokontroler AT89S8252, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal *reset* kapasitor dihubungkan dengan Vcc dan sebuah resistor yang dihubungkan ke *ground*. Rangkaian *reset* ditunjukkan dalam gambar 3-3 sebagai berikut :



**Gambar 3-3.** Perancangan Rangkaian Reset  
(Sumber : Perancangan dan Pembuatan)

Karena kristal yang digunakan mempunyai frekuensi sebesar 11,0592 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{ATAL}} = \frac{1}{11,0592 MHz} \text{ s} = 9,042 \times 10^{-8} \text{ s}$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk mereset mikrokontroler adalah :

$$\begin{aligned} \text{reset(min)} &= T \times \text{periode yang dibutuhkan} \\ &= 9,042 \times 10^{-8} \times 24 = 2,17 \mu\text{s} \end{aligned}$$

Jadi mikrokontroler membutuhkan waktu minimal 2,17  $\mu\text{s}$  untuk mereset.

Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C.

Dengan menentukan nilai R = 10 k $\Omega$  dan C = 10  $\mu\text{F}$ , maka :

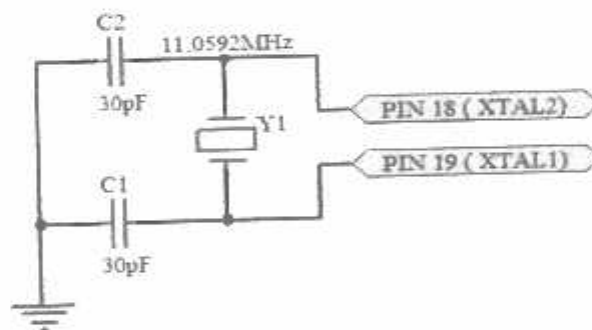
$$t = 1,1 RC = 1,1 \times 10000 \Omega \times 10 \cdot 10^{-6} = 110 \text{ ms}$$

Jadi dengan nilai komponen  $R = 10 \text{ k}\Omega$  dan  $C = 10 \mu\text{F}$  dapat memenuhi syarat minimal untuk waktu yang dibutuhkan oleh mikrokontroler.

### 3.3.2.2. Perancangan Clock

Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber *clock* yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini menggunakan osilator internal yang telah tersedia dalam *chip* AT89S8252. Untuk menentukan frekuensi osilatornya cukup dengan menghubungkan kristal dalam pin 19 ( $X_1$ ) dan pin 18 ( $X_2$ ) serta dua buah kapasitor ke *ground*.

Besarnya kapasitansinya disesuaikan dengan spesifikasi dalam lembar data AT89S8252 yaitu 30 pF. Kristal yang digunakan adalah 11,0592 MHz. Gambar 3.4 memperlihatkan rangkaian *clock* yang dirancang.



Gambar 3-4. Perancangan Rangkaian Clock

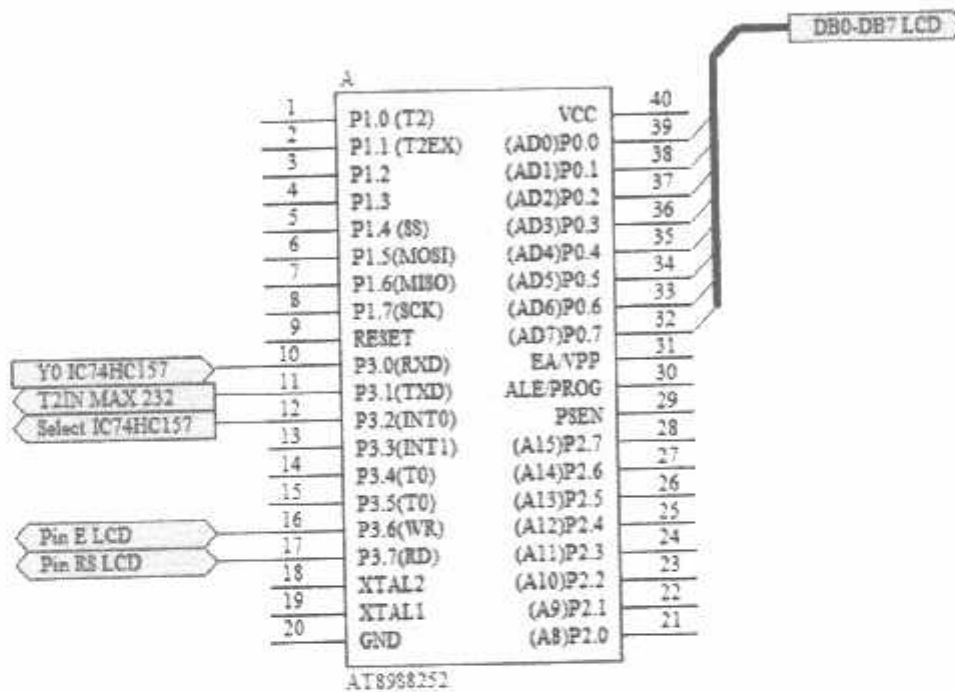
(Sumber : Perencanaan dan Pembuatan)

### 3.3.2.3. Perancangan Penggunaan Port pada Mikrokontroler AT89S8252

Dalam perancangan dan pembuatan rangkaian sistem ini menggunakan mikrokontroler AT89S8252 sebagai pengontrol masukan dan keluaran. IC ini



memiliki fasilitas EEPROM sebagai memory program sehingga dapat mempermudah dalam proses perencanaan sistem alat ini. Adapun aplikasi mikrokontroler AT89S8252 dalam perancangan ini sebagai berikut :



**Gambar 3-5.** Rangkaian MCU AT89S8252

(Sumber : Perancangan dan Pembuatan)

Agar sebuah mikrokontroler dapat bekerja sebagai pengontrol, maka kaki-kaki / port mikrokontroler dihubungkan pada rangkaian-rangkaian eksternal. Dalam perancangan ini, port yang digunakan adalah sebagai berikut :

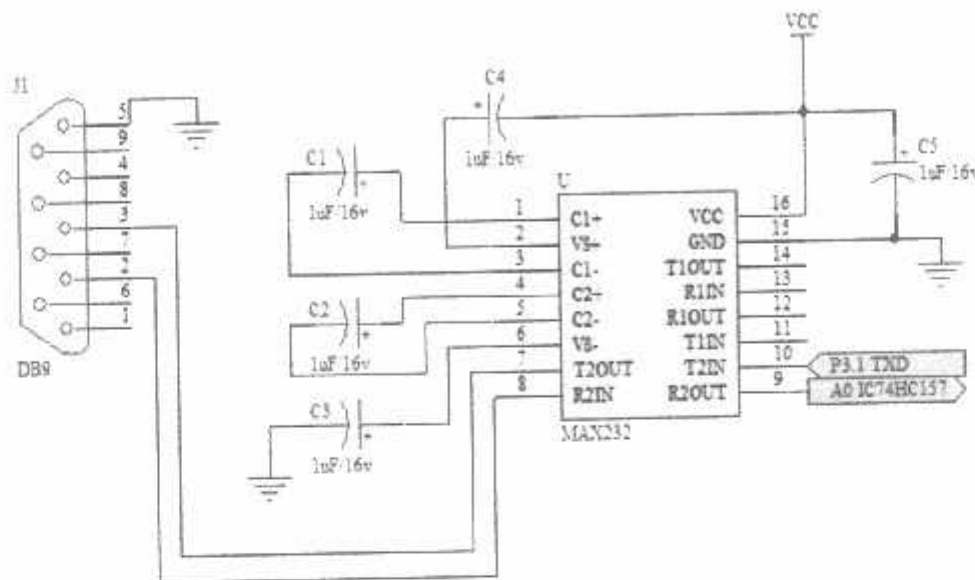
- 1) Port 0, P0.0 – P0.7 ( pin 32 s/d pin 39 ) sebagai output dari mikrokontroler yang dihubungkan ke LCD untuk menampilkan berupa informasi.

## 2) Port 3

- a) P3.0 ( pin 10 ) RXD serial input yang dihubungkan ke pin 4 ( Y0 ) dari IC 74HC157 yang digunakan untuk menerima RFID reader dan PC.
- b) P3.1 ( pin 11 ) TXD digunakan untuk port TX serial output, port ini digunakan mikrokontroler untuk mengirim data serial ke PC.
- c) P3.2 ( pin 12 ) yang dihubungkan ke pin 1 ( Select ) dari IC 74HC157 yang digunakan sebagai selektor antara RFID reader dan PC.
- d) P3.6 ( pin 16 ) digunakan sebagai port untuk pin E pada LCD.
- e) P3.7 ( pin 17 ) digunakan sebagai port untuk pin RST pada LCD.

### 3.3.3. Rangkaian Antar Muka RS-232

Pada perencanaan hubungan antara MCU dan PC menggunakan komunikasi data secara serial yang biasa dikenal dengan COM. Adapun kaki/ pin-pin yang dipakai adalah pada pin no.2 sebagai *received data*, dan pin no.3 sebagai *transmitted data* dan pin no 5 untuk *Signal ground*. Level tegangan dari RS-232 harus disesuaikan ke level tegangan TTL menggunakan IC MAX 232. Kecepatan transfer data per *bit* menggunakan 9600 bps. Rangkaian Interface RS-232 diperlihatkan pada gambar 3-6.



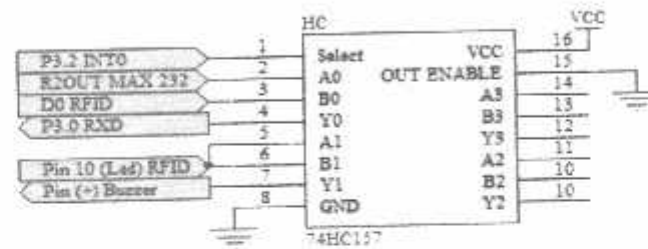
**Gambar 3-6.** Rangkaian Antar Muka RS-232  
(Sumber : Perencanaan dan Pembuatan)

Data biner yang berasal dari komputer mempunyai level tegangan antara +3 V sampai +15 V dan -3 V sampai -15 V, sebelum masuk pada rangkaian modulator dan demodulator terlebih dahulu teganganya melalui sebuah IC MAX 232, dan diubah menjadi tegangan TTL sebesar 0 Volt sampai 5 Volt yang sesuai dengan tegangan untuk mencatu kerja rangkaian modulator demodulator tersebut.

### 3.3.4. IC 74HC157A

IC ini merupakan IC multiplexer atau selektor sehingga pada perancangan sistem ini digunakan untuk memilih jalur data dari RFID ke mikrokontroler dan jalur dari PC ke mikrokontroler. Karena keduanya sama-sama membutuhkan pin RX yang ada pada mikrokontroler, sehingga harus digunakan secara bergantian. Pada perancangan ini untuk IC 74HC157 hanya menggunakan 1 output yaitu Y0 pada pin 4 dengan 2 input yang terdiri dari A0 ( pin 2 ) yang dihubungkan ke

R2OUT Max232 sedangkan dari B0 dihubungkan ke D0 RFID. Kemudian pada pin1 ( Select ) adalah sebagai pemberi logika yang nantinya akan menentukan keluaran tersebut tersambung ke Max232 atau ke RFID. Untuk Output Enable dihubungkan ke *ground* agar sistem berjalan sesuai dengan sistem alat yang direncanakan. Adapun rangkaian nya sebagai berikut :

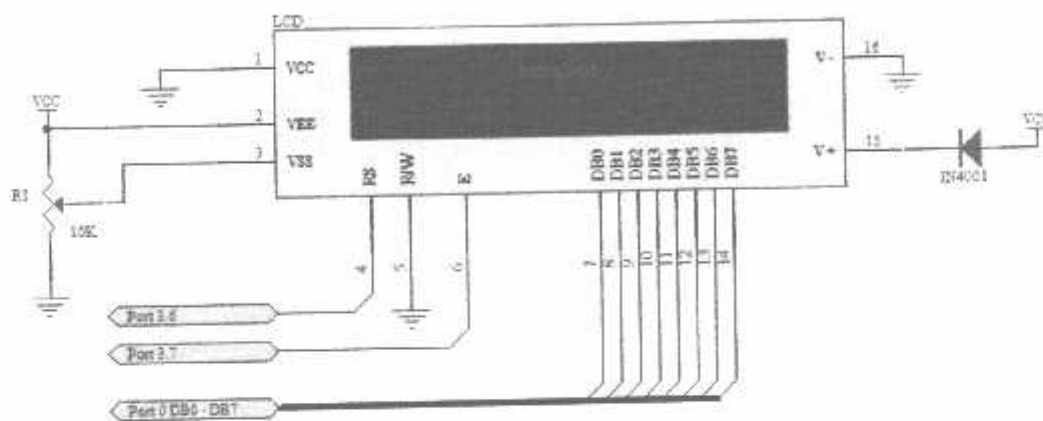


**Gambar 3-7.** Rangkaian IC 74HC157  
(Sumber : Perencanaan dan Pembuatan).

### 3.3.5. Perancangan Rangkaian LCD

Pada perancangan ini digunakan LCD Dot Matrik 2 x 16 karakter yaitu M1632. Sinyal-sinyal yang diperlukan oleh LCD adalah RS dan Enable, sinyal RS dan Enable dipergunakan sebagai input yang outputnya dipakai untuk mengaktifkan LCD. LCD akan aktif apabila mikrokontroler memberikan instruksi tulis pada LCD. Saat kondisi RS *don't care* dan Enable 0 maka LCD tetap pada kondisi semula, pengiriman data ke LCD dilakukan saat RS berlogika 0 dan enable berlogika 1. Instruksi dikirim pada LCD bila keadaan RS 1 dan Enable 1. Pin LCD ini untuk data terkoneksi pada *Port 0* mikrokontoler AT89S8252. Kemudian untuk RS dihubungkan pada *Port 3.7*, tulis/baca (*Read/Write*) diberikan logika *low* karena disini LCD bersifat menulis data, dan yang terakhir

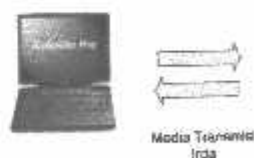
*Enable* (E) dikendalikan dengan *Port 3.6*. Gambar rangkaian LCD ditunjukkan pada gambar 3 – 10.



**Gambar 3-8.** Perancangan Rangkaian Liquid Crystal Display ( LCD )  
(Sumber : Perencanaan dan Pembuatan).

### 3.3.6. Perancangan *Interface Ponsel* dengan *PC Notebook ( Laptop )*.

Dalam perancangan sistem alat ini adalah mengirim informasi *via* SMS ( *Short Message Service* ) ke setiap orang tua / wali murid masing-masing, maka *ponsel* pada sistem ini digunakan sebagai *ponsel* operator yang semua kegiatannya dapat di akses di sebuah komputer *laptop* melalui media transmisi yaitu *port irda*. Pada perancangan sistem alat ini menggunakan *laptop Toshiba Satellite Pro 6100* dan *ponsel* Nokia 8250.



**Gambar 3-9.** Interface *Notebook* dengan *Ponsel*  
(Sumber : Perencanaan dan Pembuatan)

### 3.4. Perancangan Perangkat Lunak (*Software*)

Perangkat lunak adalah sebuah jembatan yang menghubungkan keseluruhan komponen yang ada pada sebuah komputer. Pada perancangan ini ada dua macam, yaitu perancangan perangkat lunak untuk mikrokontroler menggunakan bahasa *Assembly* Mikrokontroler standard MCS-51 dan perancangan perangkat lunak untuk komputer menggunakan *Delphi7*.

#### 3.4.1. Program Aplikasi Mikrokontroler

Program mikrokontroler bertujuan untuk mengontrol masukan dan keluaran. Hal-hal yang dikontrol mikrokontroler adalah proses komunikasi serial antara RFID dengan Komputer *Laptop*.

Didalam proses komunikasi serial antara MCU dengan *PC* terlebih dahulu ditentukan *baud rate* yang digunakan. *Baud rate* yang dibangkitkan *Timer1* dengan *Timer2* ( 8 bit *auto reload* ) yang hanya menggunakan register TH1. Pada sistem ini digunakan *baudrate* sebesar 9600 dengan SMOD = 0 dengan menggunakan Frekuensi\_ osilator = 11.0592 MHz.

$$Baud\ rate = \frac{2^{SMOD}}{32} \times \frac{Frekuensi\_osilator}{12 \times [256 - (TH1)]}$$

$$9600 = \frac{2^0}{32} \times \frac{11.059 \times 10^6}{12 \times [256 - (TH1)]}$$

$$[256 - (TH1)] = \frac{1}{32} \times \frac{11.059 \times 10^6}{12 \times 9600}$$

$$[256 - (TH1)] = 3$$

$$TH1 = 256 - 3 = 253_D = 0xFD_H$$

### 3.4.2. Program Aplikasi Komputer

Program aplikasi adalah *software* di dalam komputer yang berfungsi untuk melakukan pengendalian dengan dunia luar. Program komputer ini bertujuan untuk mengorganisasi komunikasi antara komputer sebagai pengolah data dengan mikrokontroler sebagai saluran masukan dan keluaran, sehingga perangkat lunak dalam program komputer perlu diketahui terlebih dahulu aturan-aturan atau protokol komunikasi yang digunakan untuk mengatur jalanya komunikasi antara komputer dengan mikrokontroler. Dalam perencanaan ini juga menggunakan bahasa pemrograman *visual* yaitu *Delphi7*. Komponen-komponen *Delphi7* yang digunakan untuk menunjang perancangan sistem alat dapat bekerja dengan baik, maka ada beberapa komponen tambahan yang bukan standard bawaan *Delphi7* yang juga digunakan dalam perancangan sistem alat ini.

#### 3.4.2.1. Component Pallete Delphi7

Komponen-komponen Delphi disusun di bagian *Component Pallete*, dan dikelompokkan ke dalam *Tab/page*. Komponen-komponen yang digunakan dalam perancangan sistem alat ini adalah sebagai berikut :

##### 1. Tab Additional :

Page ini berisi komponen visual pelengkap dari page standard.

##### ▪ *StringGrid*

Komponen ini untuk menampilkan data string dalam baris dan kolom.

##### ▪ *Image*

Komponen ini untuk menampilkan grafik seperti icon, bitmap, metafile.

▪ **Bevel**

Komponen ini untuk menggambar segiempat dengan tampilan insert atau menonjol.

**2. Tab Standard :**

Sesuai dengan namanya page standard ini berisi komponen yang diperlukan untuk membangun aplikasi Windows yang standard. Bagian ini berisikan komponen visual maupun nonvisual yang terdiri dari empat belas komponen, namun dalam perencanaan ini hanya menggunakan beberapa komponen saja yaitu:

▪ **Label**

Komponen ini digunakan untuk membuat teks di form atau obyek lain tanpa dapat diubah oleh pemakaian program (VC).

▪ **Memo**

Komponen ini dipakai untuk menerima masukan/ menampilkan beberapa baris teks.

▪ **Panel**

Komponen ini merupakan kontainer untuk mengelompokkan komponen-komponen, misalnya Label, Edit, Status Bar dsb (VC).

**3. Tab Data Access**

Page ini semuanya berisi nonvisual yang mengakses sumber data misalnya database, seperti Paradox, MS SQL, MS Access juga dapat berhubungan lewat ODBC.



#### ▪ *Data Source*

Data Source bertindak sebagai penghubung antara komponen pengakses data dengan DataSet.

#### 4. Tab Dialogs

Page ini berisi komponen non visual untuk pengaksesan kotak dialogs dari Windows.

#### ▪ *OpenPictureDialog*

Komponen ini untuk pengaksesan kotak dialogs yang berisikan gambar.

#### 5. Tab Samples

Untuk memperlihatkan komponen yang dapat ditulis.

#### ▪ *SpinEdit*

Komponen ini dapat digunakan untuk mengedit nilai/ value pada saat dalam kondisi run (VC).

#### 6. Tab SUIPack

Dalam perencanaan ini juga menggunakan komponen tambahan yang tidak ada dalam standard komponen yang dimiliki oleh delphi. Komponen ini adalah SUIPack, komponen ini gratis untuk di download sehingga memudahkan para penggunanya. Komponen SUIPack ini sebenarnya memiliki fungsi yang sama dengan komponen standard bawaan delphi namun komponen SUIPack ini adalah komponen Skin yang tampilannya lebih bagus dan lebih menarik. Sehingga kita membuat program yang bagus hanya mengandalkan komponen dengan *source code* yang lebih minim.

---

#### ▪ *suiForm*

Sebuah jendela kosong yang digunakan untuk merancang aplikasi Windows, untuk menempatkan komponen, kontrol, dan lain-lain.

#### ▪ *suiFileTheme*

Sebuah themefile yang nantinya bisa digunakan sebagai tampilan untuk komponen lainnya, misalnya Panel, Button, ComboBox, dan lain-lain.

#### ▪ *suiToolBar*

Dengan suitoolbar dapat melakukan beberapa operasi pada menu utama yang setiap tombol berisi informasi mengenai fungsi dari tombol tersebut.

#### ▪ *suiButton*

Untuk membuat tombol dengan beberapa pilihan style.

#### ▪ *suiPanel*

Komponen ini merupakan kontainer untuk mengelompokkan komponen-komponen, misalnya Label, Edit, Status Bar dsb (VC). Dengan beberapa pilihan style.

#### ▪ *suiComboBox*

Komponen ini sama seperti ListBox, namun *suiComboBox* ini memiliki unsur komponen edit. User dapat memilih item dan dapat mengetikkan teks ke dalam kotak. Style komponen ini ada beberapa pilihan yang dapat digunakan.

#### ▪ *suiProgressBar*

*SuiProgresBar* digunakan untuk mengetahui progres dari jalanya prosedur. Progres akan terisi dari kiri ke kanan. Dengan fasilitas style pilihan.

---

## 7. Tab SUIPackDB

Komponen ini juga sama seperti komponen databse bawaan Delphi. Namun komponen ini memiliki skin yang lebih menarik.

- *suiDBGrid*

Merupakan tampilan dan edit Dataset dalam format tabular (baris dan kolom).

- *suiDBNavigator*

Untuk mengontrol/ menggerakkan posisi rekord, seperti Preveious, Next, First, Last Record dan mengubah statusnya seperti Open, Close, Edit, Post (simpan), Delete, Cancel edit.

- *suiDBEdit*

Menampilkan dan mengedit sebuah field dalam kotak edit 1 baris.

- *suiCheckBox*

Untuk memilih atau membatalkan pilihan, caranya dengan mengklik pada checkBox.

## 8. Tab AM Library

Komponen ini tidak ada dalam bawaan Delphi. Komponen ini mencakup komponen visual dan nonvisual.

- *AMDBF*

Komponen yang *stand alone* untuk database *Engine* yang hanya *support* database IV atau disebut DBF IV yang diserahkan penuh dalam *borland*.

- *packetComPort*

Komponen yang khusus untuk menghendel serial komunikasi.

---

- ***amProtector***

Komponen *link* untuk bisa mengaktifkan AMDBF dan pocketComport.

- ***amWaitingGradient***

Komponen visual untuk dengan tampilan bergeser ke arah kanan ke kiri.

Dalam perancangan ini digunakan untuk status *scan hardware*.

- ***hiTimer***

Untuk mengaktifkan procedure, function dan event pada interval waktu tertentu.

- ***scrollingCaption***

Menampilkan caption yang bisa berjalan dari kiri atau dari kanan. Contoh penggunaanya pada Form.

- ***amEnterAsTab***

Untuk database komunikasi operasional, yaitu mengubah tomhol TAB menjadi ENTER untuk berpindah fokus dari satu kontrol ke kontrol yang lain.

## 9. Tab Oxygen

Komponen ini terdiri hanya satu komponen nonvisual.

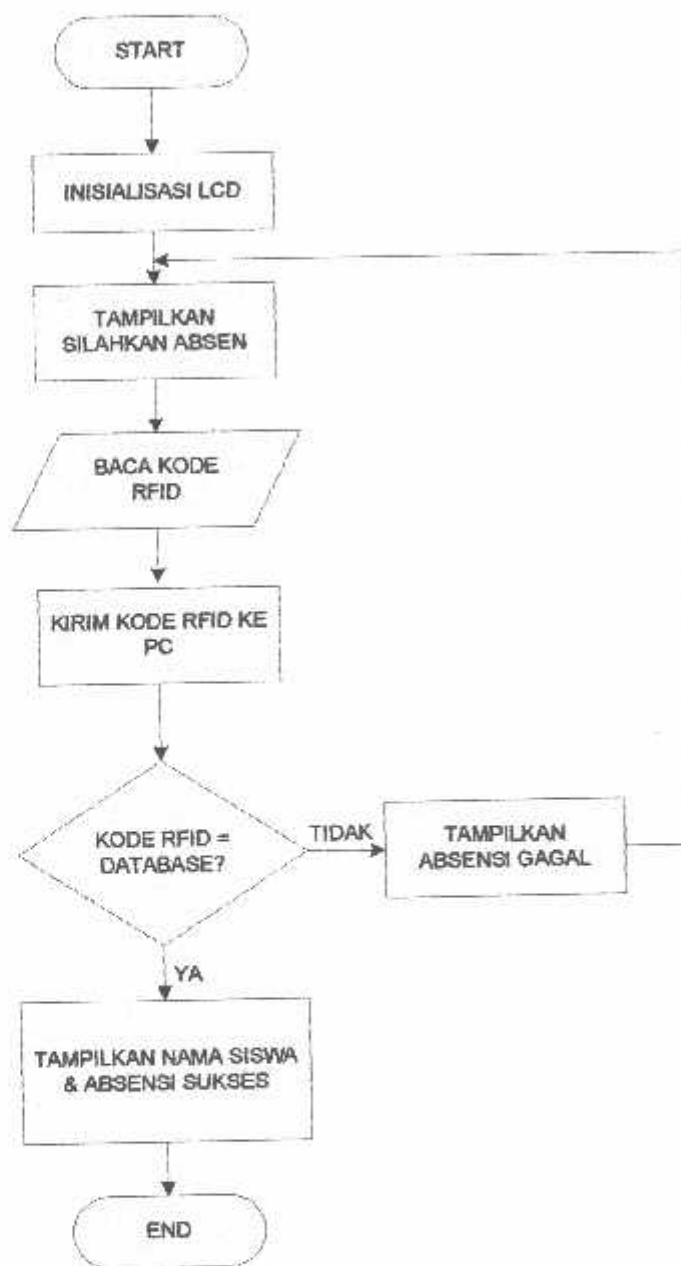
- ***OxygenSMS***

Komponen ini digunakan untuk menghubungkan HP dengan komputer. Sebelum adanya komponen yang dimiliki Delphi ini, para pengguna masih menggunakan *AT Command* untuk aplikasi SMS. Namun dengan komponen delphi ini lebih ringkas tidak banyak listing dan lebih efisien. Untuk sekarang ini penggunaan *TOxygenSMS* hanya bisa diakses untuk HP tertentu saja.

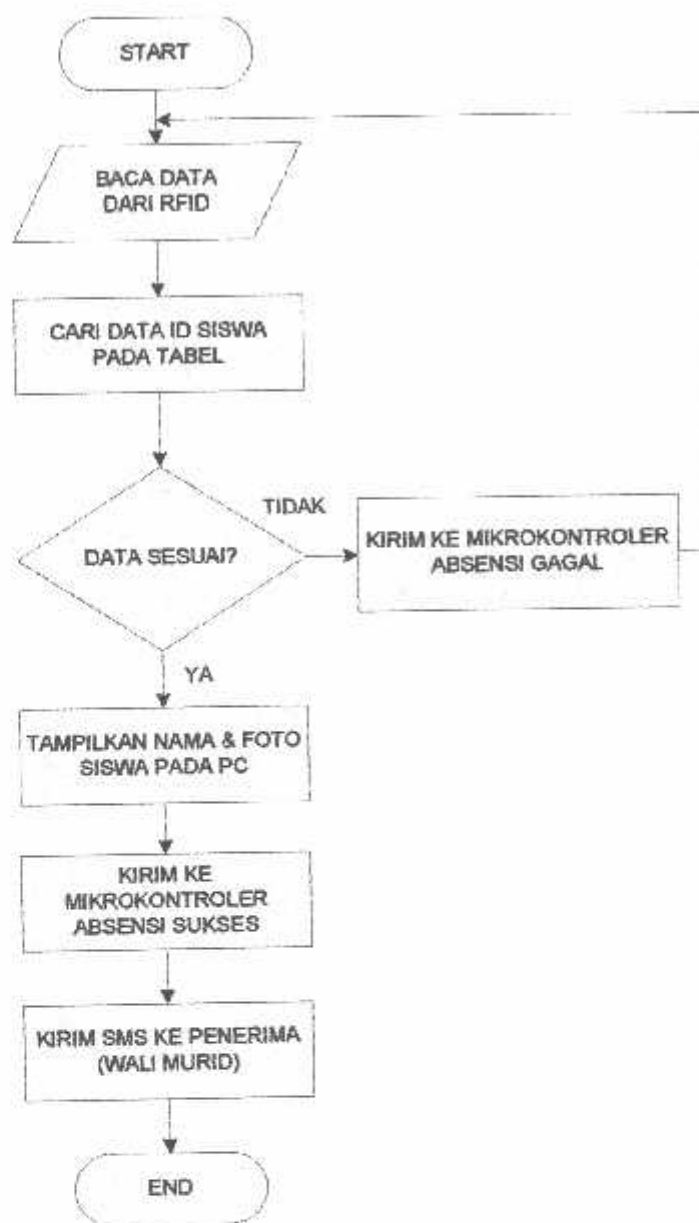
---

Pada perancangan sistem alat ini, sesuai dengan rencana sistem alat ini bahwa media transmisi menggunakan *infrared* maka harus mengeset properties yang ada pada komponen *TOxygenSMS* ( sesuai dengan yang sudah dijelaskan pada bab 2 ). Bahwa untuk *infrared ConnectionMode* di set 2.

## 3.4.3. FlowChart



**Gambar 3-10.**  
Diagram Alir Program Mikrokontroler



**Gambar 3-11.**  
Diagram Alir Program Komputer

## BAB IV

### ANALISIS DAN PENGUJIAN ALAT

#### 4.1. Pendahuluan

Dalam bab ini membahas tentang pengujian dan pengukuran dari peralatan yang dibuat. Secara umum pengujian ini bertujuan untuk mengetahui apakah piranti yang telah direalisasikan dapat bekerja sesuai dengan perencanaan yang telah direncanakan.

#### 4.2 Pengujian RFID

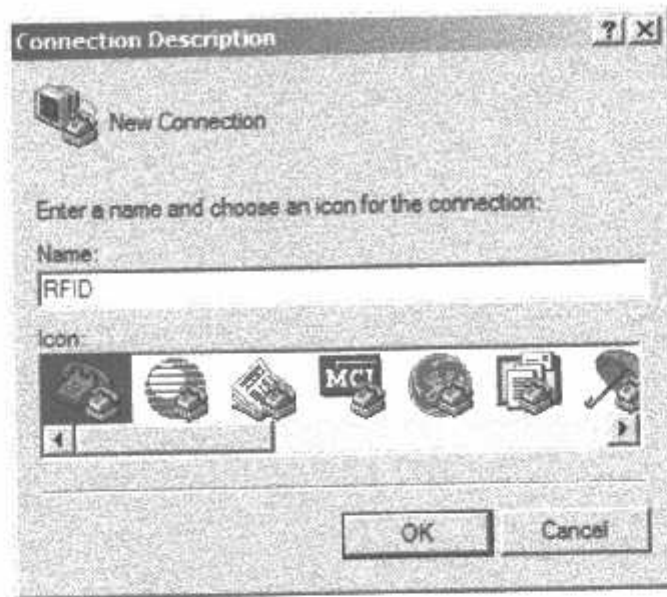
##### 4.2.1. Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui *tag* RFID bisa dibaca oleh *reader* RFID. Adapun cara pengujianya adalah dengan merangkai rangkaian RFID dan kemudian menghubungkan ke COM 1 PC. Untuk menguji *reader* bisa membaca kartu RFID dilakukan melalui *Hyper Terminal*.

##### 4.2.2. Prosedur Pengujian

- a. Menghubungkan rangkaian RFID ke COM 1 PC.
  - b. Membuka *Hyper Terminal* (*Start* → *all program* → *accessories* → *Communication* → *hyperterminal* ).
  - c. Memberi nama dan memilih *icon* pada *Connection Desert*.
-





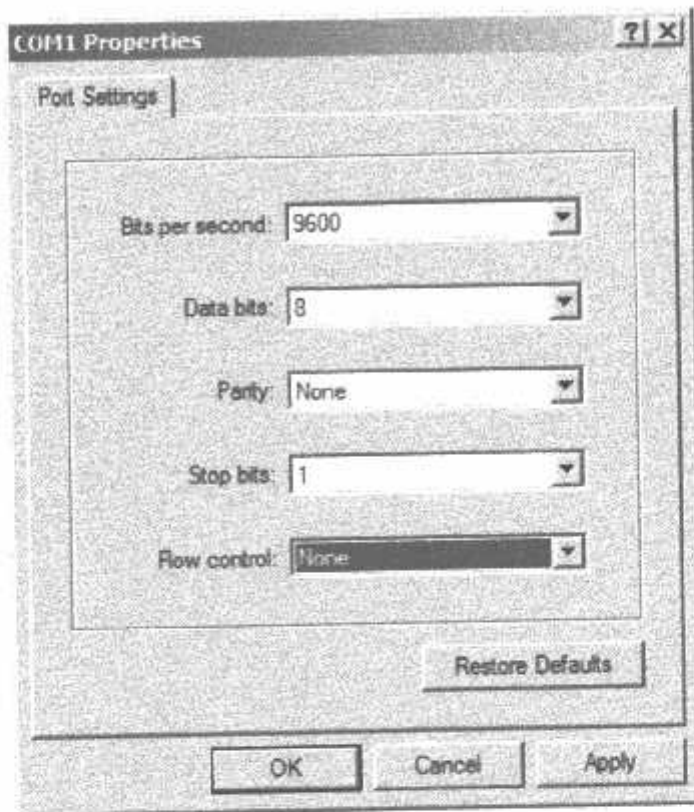
**Gambar 4 - 1.** Kotak Dialog Connection Description.

- d. Memilih COM 1 pada kotak dialog *connect to*.



**Gamabar 4 - 2.** Kotak Dialog Connect To.

- e. Pada COM 1 *properties* mengubah *bits, rate per second* menjadi 9600 dan *flow control* menjadi *none*.



Gambar 4 - 3. Kotak Dialog COM 1 Properties

- f. Menempatkan kartu pada jarak yang dijangkau *reader* sehingga menampilkan angka dari kartu tersebut.



Gambar 4 - 4. Identifikasi Reader Terhadap kartu.

4.2.3. Hasil Pengujian Pembacaan RFID

Tabel 4 - 1. Hasil Pengujian Pembacaan RFID

Jarak	Percobaan									
	1	2	3	4	5	6	7	8	9	10
1 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6 cm	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
7 cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
8 cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
9 cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
10cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
11 cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
12 cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Tabel diatas merupakan hasil pengujian dimana kartu yang menghadap *reader* adalah bagian beakang. Jarak yang baik untuk bisa teridentifikasi adalah 5 cm. Untuk bagian depan menghasilkan data yang sama, tetapi untuk pengujian dimana kartu tegak lurus dengan *reader* hanya bisa saat kartu berjarak sangat dekat dengan *reader* ( menempel ).



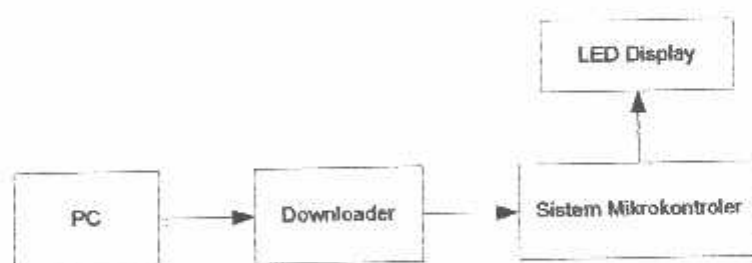
**Gambar 4 - 5.** Jarak Pembacaan Reader Terhadap kartu.

### 4.3. Pengujian Mikrokontroler AT89S8252

#### 4.3.1. Tujuan

Untuk mengetahui kondisi awal dari Mikrokontroler apakah sudah sesuai yang direncanakan.

#### 4.3.2. Prosedur Pengujian



**Gambar 4 - 6.** Diagram Blok Pengujian Mikrokontroler.

1. Rangkaian dibuat seperti gambar 4 – 6.
2. Memberikan catu daya 5 volt.
3. Membuat program yang akan digunakan dalam pengujian mikrokontroler.

Program yang akan digunakan dalam pengujian mikrokontroler ini merupakan program yang sederhana yang meletakkan 0F<sub>16</sub> dan F0<sub>16</sub> secara bergantian pada Port 2 AT89S8252. Program yang dibuat adalah sebagai berikut :

```
ORG 00H

START:  MOV P2, #0FH
        CALL TUNDA
        MOV P2, #0F0H
        CALL TUNDA
        JMP START

TUNDA:  MOV R1, #8
LOOP1:  MOV R2, #250
LOOP2:  MOV R3, #250
LOOP3:  DJNZ R3, LOOP3
        DJNZ R2, LOOP2
        DJNZ R1, LOOP1
        RET
        END
```

4. Download Program diatas.
5. Mengamati keluaran pada LED Display.

#### 4.3.3. Hasil Pengujian

Tabel 4 - 2. Hasil Pengujian Sistem Mikrokontroler

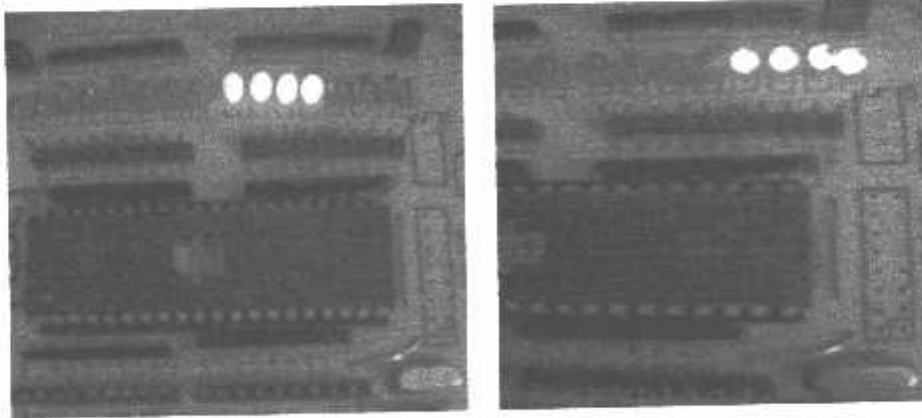
Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Satu	0	0	0	0	1	1	1	1
Dua	1	1	1	1	0	0	0	0

Keterangan :

- Kondisi Bit Low ( 0 ) = Led Menyala
- Kondisi Bit High ( 1 ) = Led Mati

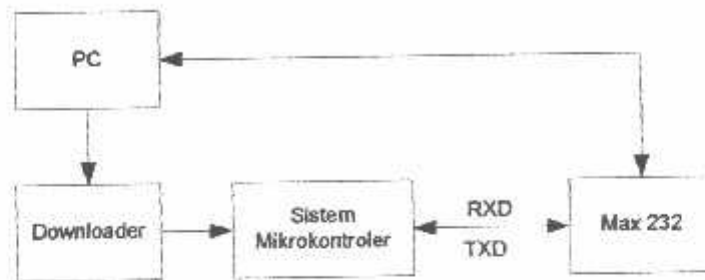
#### 4.3.4. Analisis Hasil Pengujian

Dari hasil pengujian dalam tabel 4 – 2 dapat dilihat bahwa *port 2* memberikan logika  $0F_H$  seperti gambar sebelah kiri dan  $F0_H$  pada gambar sebelah kanan secara bergantian sesuai dengan isi program.



Gambar 4-7. Hasil Pengujian Tampilan Led

#### 4.4. Pengujian Serial 232



**Gambar 4 - 8.** Rangkaian Pengujian Transfer Data.

##### 4.4.1. Tujuan

Untuk mengetahui apakah data yang dikirim dari MCU ke PC dapat diterima dengan benar dengan melakukan simulasi pada komputer.

##### 4.4.2. Peralatan Yang Digunakan

- ❖ Komputer
- ❖ Sistem Mikrokontroler dengan antarmuka RS232

##### 4.4.3. Prosedur Pengujian

1. Menyusun rangkaian seperti gambar 4-8.
  2. Membuat program transfer data pada sistem mikrokontroler seperti yang ditunjukkan dalam gambar 4-8 .
  3. Downloads program ke Mikrokontroler dan eksekusi program
  4. Membuat hasil yang terlihat dalam layar komputer.
- ❖ Program Uji Komunikasi serial dari Mikrokontroler

ORG 00H

START : CALL INIT

```
LOOP : CALL RECV_SR
      CJNE A, 31H, DUA
      CPL P2.0
      CALL SEND_SR
      JMP LOOP

DUA : CJNE A, 32#H, LOOP
      CPL P2.1
      CALL SEND_SR
      JMP LOOP

INIT:
      MOV TMOD,#20H
      MOV TCON,#41H
      MOV TH1,#0FDH
      MOV PCON,#00H
      MOV SCON,#50H
      SETB TR1
      RET

SEND_SR:
      MOV SBUF,A
      JNB TI,$
      CLR TI
      RET
```

---



```
RCV_SR:
    JNB R1,SERIAL_IN
    MOV A,SBUF
    CLR,R1
    RET
END
```

Prinsip kerja program diatas adalah apabila kita mengetikan angka 1 pada *keyboard* komputer maka akan mempengaruhi kondisi Port 2 mikro pin ke-0 (P2.0), sebaliknya jika kita menekan angka 2 pada *keyboard* maka pada Port 2 mikro pin ke-1 (P2.1).

❖ Program Uji Komunikasi serial di Komputer

“ Cuplikan program uji komunikasi serial di komputer menggunakan program Delphi 7.0 “

```
Procedure TForm1.FormCreate(Sender:Tobject);
```

```
Begin
```

```
    PacketComport1.Connected:=True;
```

```
End;
```

```
Procedure Tbutton1Click(Sender:Tobject);
```

```
Begin
```

```
    PacketComport1.WriteStr('1');
```

```
End;
```

```
Procedure Tbutton2Click(Sender:Tobject);
```

```
Begin
```

```
PacketComport1.WriteStr('2');
```

```
End;
```

#### 4.4.4. Hasil Pengujian

**Tabel 4 - 3.** Hasil Pengujian Komunikasi serial.

Data yang dikirim dari Delphi	Data yang diterima Mikrokontroler
PacketComport1.WriteStr('1')	Mempengaruhi kondisi mikro Pin (P2.0)
PacketComport1.WriteStr('2')	Mempengaruhi kondisi mikro Pin (P2.1)

### 4.5. Pengujian Rangkaian Tampilan LCD

#### 4.5.1. Tujuan

Untuk mengetahui kemampuan rangkaian tampilan yang sudah dibuat apakah dapat mendukung sistem yang direncanakan dan untuk menampilkan data pada LCD.

#### 4.5.2. Peralatan yang dibutuhkan

1. Komputer ( PC )
2. Sistem Mikrokontroler dan LCD

#### 4.5.3. Prosedur Pengujian



**Gambar 4 - 9.** Diagram Blok Pengujian Rangkaian LCD.

1. Menyusun rangkaian seperti gambar 4 - 9.

2. Menjalankan program untuk menampilkan tulisan ke LCD.
3. Mengamati keluaran pada LCD.
4. Isi memori program seperti dibawah ini yang menampilkan tulisan  
"SISTEM ABSENSI RFID VIA SMS"

❖ Program Pengujian LCD

ORG 00H

```
MULAI : CALL INIS
        MOV DPTR,#BARIS_A
        MOV R3,#16
        MOV A,#80H
        CALL TULIS_INST

TULIS : CLR A
        MOVC A,@A+DPTR
        INC DPTR
        CALL TULIS_DATA
        DJNZ R3,TULIS
        MOV DPTR,#BARIS_B
        MOV R3,#16
        MOV A,#0C0H
        CALL TULIS_INST

TULIS1 : CLR A
        MOVC A,@A+DPTR
        INC DPTR
```

```
CALL TULIS_DATA
DJNZ R3,TULIS1
JMP $
INIS      : MOV A,#03FH
CALL TULIS_INST
MOV A,#0DH
CALL TULIS_INST
MOV A,#01H
CALL TULIS_INST
TULIS_INST : CLR P2.6
MOV P0,A
SETB P2.7
CLR P2.7
CALL TUNDA
RET
TULIS_DATA : SETB P2.6
MOV P0,A
SETB P2.7
CLR P2.7
CALL TUNDA
RET
TUNDA      : MOV R0,#0FFH
TUNDA1     : MOV R1,#50H
```

---

```
DJNZ R1,$  
DJNZ R0,TUNDA1  
RET  
  
BARIS_A : DB ' SISTEM ABSENSI '  
BARIS_B : DB ' RFID VIA SMS '  
  
END
```

#### 4.5.4. Hasil Pengujian

Setelah data diolah mikrokontroler maka hasil tampilan LCD berupa tulisan “  
SISTEM ABSENSI RFID VIA SMS “.



**Gambar 4-10.** Hasil Pengujian Tampilan LCD.

#### 4.6. Pengujian Buzzer

##### 4.6.1. Tujuan

Untuk mendeteksi apakah data yang dikirimkan oleh *Tag* sudah diterima oleh *reader* RFID.

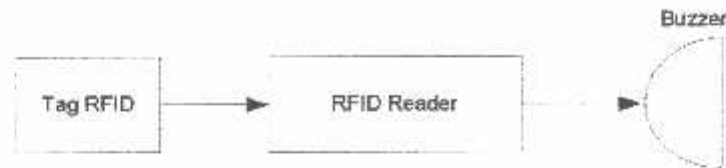
##### 4.6.2. Peralatan yang digunakan

1. *Buzzer*
2. *Tag* RFID
3. *Reader* RFID

##### 4.6.3. Prosedur Pengujian

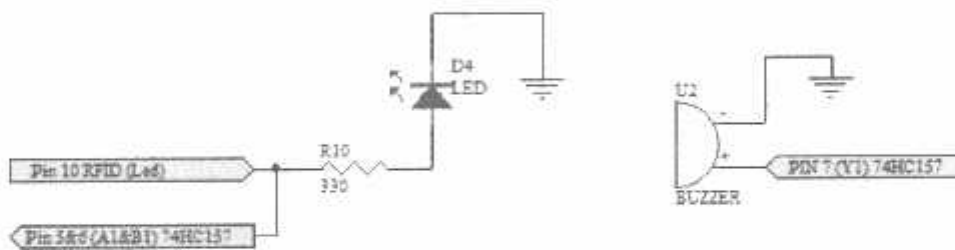
1. Menyusun rangkaian seperti pada gambar 4 – 11.

2. Mendekatkan *Tag* ke *Reader* RFID
3. Mengamati respon *buzzer*



**Gambar 4 – 11.** Blok Rangkaian *Buzzer*.

Pada perencanaan alat ini *buzzer* dihubungkan dengan pin 10 pada RFID *reader* seperti pada gambar 4 - 10.



**Gambar 4 – 12.** Rangkaian Driver *Buzzer* dan Led.

Pada perencanaan rangkaian driver diatas digunakan untuk mengaktifkan *buzzer* dan led, dimana jika kartu *tag* didekatkan dengan *reader* RFID maka *buzzer* akan berbunyi yang mengidentifikasikan bahwa *buzzer* telah bekerja.

Kemampuan arus maksimal *buzzer* 17 mA, maka dengan menggunakan IC 74HCT157A yang memiliki arus keluaran maksimal 25mA bisa untuk mengaktifkan *buzzer*. Kemudian untuk arus led dibutuhkan nilai-nilai resistor yang menghasilkan arus yang besarnya tidak melebihi kemampuan arus maksimal led. Untuk menentukan nilai resistor yang akan dipakai, menggunakan rumus pembagi tegangan seperti berikut :

$$V_{LED} = 2,4 \text{ V}$$

$$R_{LED} = \frac{V_{RFID} - V_{LED}}{I_{LED}} = \frac{5V - 2,4}{10mA} = 260 \Omega$$

Karena dipasaran resistor 260  $\Omega$  tidak ada maka diganti dengan resistor 330  $\Omega$ .

#### 4.7. Pengujian Sistem Absensi Berbasis RFID Yang Di informasikan Via SMS

##### 4.7.1. Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui apakah semua sistem berjalan dengan normal dan juga untuk mengetahui *error* yang terjadi.

##### 4.7.2. Prosedur Pengujian

1. Menghubungkan keseluruhan rangkaian sesuai dengan diagram blok
2. Menjalankan program Database Delphi
3. Melakukan proses identifikasi
4. Melewati *Tag* RFID
5. Melakukan Absensi
6. Mengirimkan SMS berupa Informasi Data Absensi

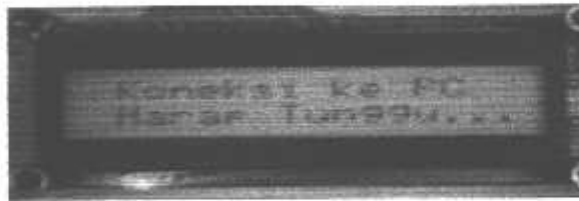
##### 4.7.3. Hasil Pengujian

1. Tampilan Informasi Pada LCD
  - a. Menampilkan Tulisan “ Koneksi Ke PC Harap Tunggu “.
  - b. Menampilkan Tulisan “ Koneksi Ke PC Terkoneksi “.
  - c. Menampilkan Tulisan “ Silahkan Absent ”.

- d. Menampilkan Tulisan “ Memproses Data Harap Tunggu “.
  - e. Menampilkan Tulisan “ Nama Siswa Absensi Sukses “.
  - f. Menampilkan Tulisan “ Kartu Anda Tidak Terdaftar “.
2. Tampilan Informasi Pada Program Delphi
- a. Menampilkan Form Absensi Automatis
  - b. Menampilkan Form Tanggal
  - c. Menampilkan Form Mencari Absensi Siswa
  - d. Menampilkan Form Laporan Absensi Siswa

#### 4.7.4. Proses Identifikasi dan Sistem Absensi

- a. Sebelum terjadi koneksi dengan PC atau Laptop, LCD akan menampilkan tulisan seperti gambar 4-13.



**Gambar 4 – 13.** Tampilan LCD Koneksi ke PC Harap Tunggu.

- b. Setelah semua perangkat di hubungkan maka LCD akan menampilkan tulisan seperti gambar 4-14 dan gambar 4-15 . Kemudian melakukan koneksi HP dengan PC atau Laptop melalui *infrared* pada Monitor akan tampil seperti gambar 4-16.

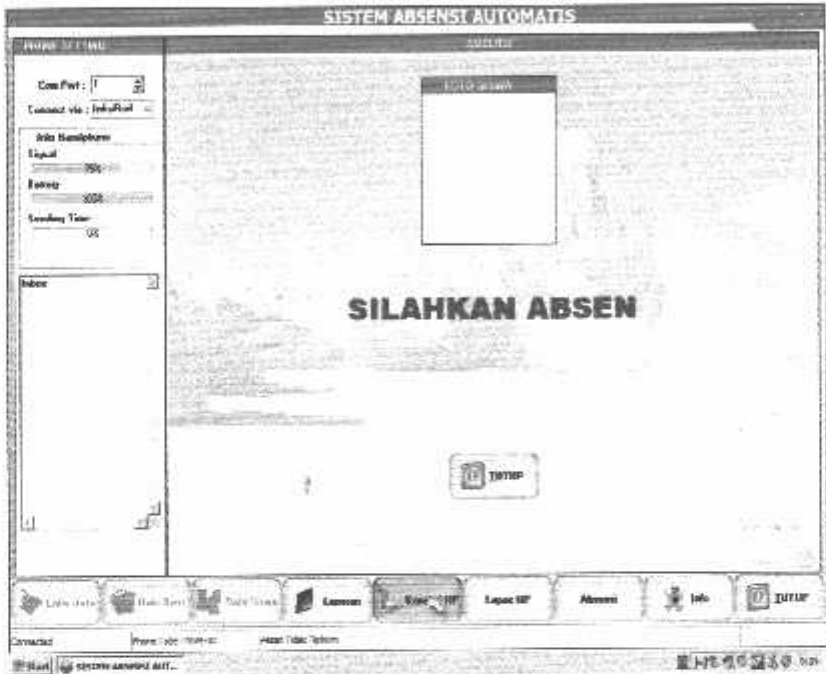


**Gambar 4 – 14.** Tampilan LCD Komunikasi ke PC Terkoneksi.





Gambar 4 – 15. Tampilan LCD Silahkan Absen.



Gambar 4 – 16. Tampilan Layar Monitor Silahkan Absen.

- c. Pada saat melakukan proses absensi atau identifikasi maka LCD akan menampilkan tulisan seperti gambar 4-17. Proses ini melakukan pemrosesan data yang telah diidentifikasi.



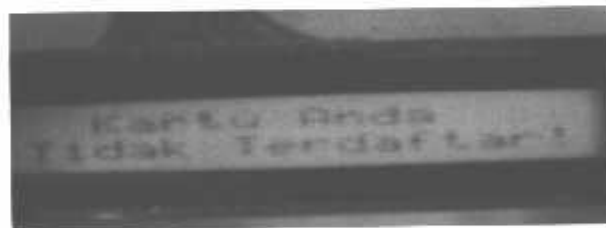
Gambar 4 – 17. Tampilan LCD Memproses Data Harap Tunggu.

IIP yang sudah ada dalam database, contoh pesan tersebut berisi “ Siswa Hansah Danuarta Masuk Sekolah Jam 10:18:34 “.



**Gambar 4 – 20.** Tampilan Pesan Informasi Siswa Masuk Sekolah.

- f. Jika data yang diidentifikasi tidak terdaftar dalam database maka LCD akan menampilkan seperti gambar 4-21. Pada layar monitor juga akan menampilkan seperti gambar 4-22.



**Gambar 4 – 21.** Tampilan LCD Data Tidak Terdaftar.



Gambar 4 – 22. Tampilan Monitor Kartu Tidak Terdaftar.

- g. Kemudian pada Form Laporan Absensi Siswa menekan tombol kirim dengan maksud bahwa HP akan mengirim pesan bagi siswa yang tidak masuk ke no HP yang sudah ada dalam database seperti gambar 4-23.. Contoh Pesan tersebut seperti gambar 4-24.

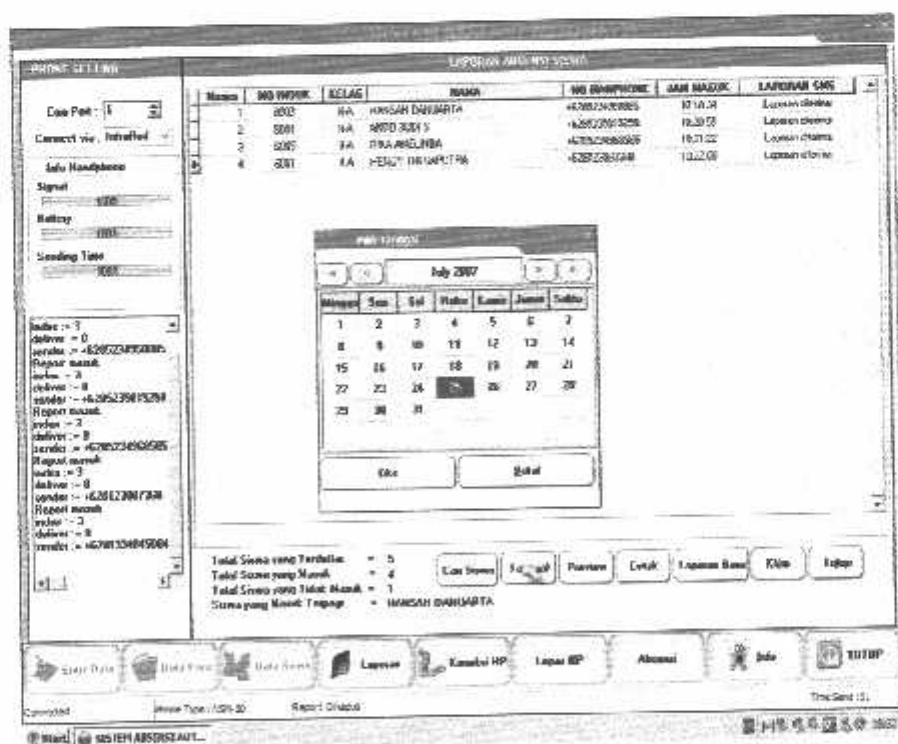


Gambar 4 – 23. Mengirim Laporan Untuk Siswa Tidak Masuk.



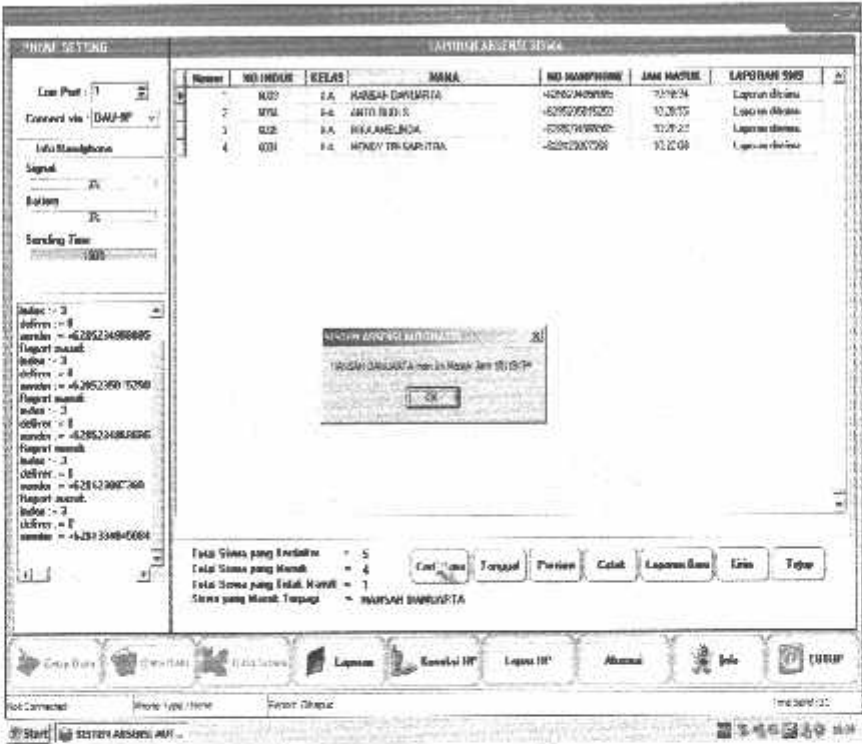
**Gambar 4 – 24.** Tampilan Pesan Informasi Siswa Tidak Masuk Sekolah.

h. Untuk mengetahui laporan hasil absensi sesuai tanggal maka bisa dilihat pada Form Tanggal seperti gambar 4-25..



**Gambar 4 – 25. Tampilan Melihat Laporan sesuai Tanggal.**

- i. Untuk mengetahui siswa tersebut telah absen atau tidak maka bisa menggunakan Form mencari absensi siswa seperti gambar 4-26.



Gambar 4 – 26. Tampilan Mencari Informasi Siswa .

- j. Preview hasil laporan dapat dilihat seperti gambar 4-27.



Gambar 4 – 27. Tampilan Preview Hasil Laporan.



```
$Include (89S8252.MCU)
$Include(Systems.MOL)
$Include(lcd.MOL)
$Include(Serial8252.MOL)
CSeg At 550h
```

```

;-----;
; user Mapping bit & memory ;
;-----;
```

```
SerialSelect      Bit   P3.2
```

```

;-----;
; LCD Redeclare ;
;-----;
```

```
LCDRS      Bit   P3.7
LCDENABLE  Bit   P3.6
```

```
CSeg
```

```

;-----;
; For User Listing Program ;
;-----;
```

```
SelectToPC:
```

```
    Clr   SerialSelect
    Call  Delay
    Call  Delay
    Mov   DPTR,#TextResume
    Call  SerialWriteString
    Ret
```

```
SelectToRFID:
```

```
    Mov   DPTR,#TextPause
    Call  SerialWriteString
    Call  Delay
    Setb  SerialSelect
    Call  Delay
    Ret
```

```
SendTagCode:
```

```
    Mov   DPTR,#TextTAG
    Call  SerialWriteString
    Mov   SerialBufferLength,#BufferSize
    Call  serialWriteBuffer
    Ret
```

---

LogoKampus:

```

    Call  LCDClearDisplay
    Mov   DPTR,#TextJudul
    Call  LCDDisplayMessage
    Mov   DPTR,#TextJudul2
    Call  LCDDisplayMessage

    Mov   DPTR,#TextNama
    Call  LCDDisplayMessage

    Mov   DPTR,#textNim
    Call  LCDDisplayMessage

    Mov   DPTR,#TextUniversitas
    Call  LCDDisplayMessage
    Ret

```

KONEKSIPC:

```

    Mov   DPTR,#TextOnLine
    Call  SerialWriteString

```

WaitLinkToPC:

```

    Call  SerialRead
    Call  SerialWrite
    Cjne  A,#8EH,DoLinks
    Call  SelectToPC
    Mov   DPTR,#TextConectedOK
    Call  LCDDisplayMessage
    Mov   DPTR,#TextResume
    Call  SerialWriteString
    Jmp   EndWait

```

DoLinks:

```

    Cjne  A,#0E5H,UnlinkPC
    Mov   DPTR,#TextOnLine
    Call  SerialWriteString
    Jmp   WaitLinkToPC

```

UnlinkPC:

```

    Cjne  A,#8FH,WaitLinkToPC
    Mov   DPTR,#TextConected
    Call  LCDDisplayStr
    Call  SelectToPC
    Jmp   WaitLinkToPC

```

EndWait:

```

    RET

```



MAIN:

```

Call LCDInit
Call SerialInit

Call LogoKampus
Call SelectToPC

Mov DPTR,#TextConected
Call LCDDisplayMessage
Setb EA
Call KONEKSIPC

Mov DPTR,#TextGoAbsen
Call LCDDisplayMessage

```

Proses Utama
--------------

Preloop:

```

Jb SerialSelect,RFIDHandle
Call serialReadHandle
Jc Preloop

```

StartStr:

```

Cjne A,#0E1h,PrintOut
Call SerialFirstBuffer
Jmp endDTCase

```

PrintOut:

```

Cjne A,#0E2h,ValidTag
Mov A,#10h
Call LCDAtXY
Mov R0,#SerialBufferSet
Mov R6,#16
Call LCDWriteRam
Mov DPTR,#TextAbsenOK
Call LCDWriteStrOnLine2
Jmp endDtCase

```

ValidTag:

```

Cjne A,#0E3H,InvalidTag
Mov DPTR,#TextAbsenOK
Call LCDWriteStrOnLine2
Jmp endDtCase

```

InvalidTag:

```

Cjne A,#0E4H,READTAG
Mov DPTR,#TextNotReg

```

```

    Call  LCDDisplaySTR
    Jmp   EndDtCase

ReadTag:
    Cjne  A,#0E6H,ENDCASE
    Mov   SerialBufferLength,#00
    Call  SelectToRFID
    Jmp   EndDtCase

EndDtCase:
    Jmp   Preloop

RFIDHandle:
    Mov   SerialBufferLength,#16
    Call  SerialReadBuffer

    Call  SendTagCode
    Mov   DPTR,#TextProses
    Call  LCDDisplayStr
    Call  SelectToPC
    Jmp   Preloop
    Ljmp  MAIN

TextJudul:      Db ' SISTEM ABSENSI $ MURID SEKOLAH $'
TextJudul2:     Db ' MENGGUNAKAN $ RFID VIA SMS $'
TextMain:       Db ' SISTEM ABSENSI $ OTOMATIS $'
TextNama:       Db ' LISTYAMI $ FATMAWATI $'
TextNim:         Db ' NIM $ 03.17.110 $'
TextUniversitas: Db ' ELEKTRONIKA SI $ ITN MALANG $'

TextConected:   Db ' Koneksi ke PC $ Harap Tunggu... $'
TextConectedOK: Db ' Komunikasi ke PC $ * Terkoneksi * $'
TextTAG:        Db ' TAG=$'
TextOnLine:     Db 'ON LINE$'
TextPause:      Db 'PC Pause$'
TextResume:     Db 'PC Resume$'
TextProses:     Db ' Memproses Data $ Harap Tunggu.. $'

TextAbsenOK:    Db ' Absen sukses $'
TextGoAbsen:    Db ' Silahkan $ Absen $'
TextNotReg:     Db ' Kartu Anda $ Tidak Terdaftar! $'

End

```

```
unit MAIN;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, Jpeg,
ExtDlgs, Buttons, Spin, ComCtrls, ExtCtrls, Grids, DBGrids, DBCtrls, Mask, DB,
StdCtrls, Menus, FileCtrl, SUIMgr, SUIForm, SUIButton, ToolWin,
SUIToolBar, SUIStatusBar, SUIImagePanel, SUIDBCtrls, SUIComboBox,
SUIProgressBar,
SMSComp, AmWaitingGradient, ThdTimer, ScrIcapt, AmEnterTab,
AMComPorts, Amdbf, amProtector;
```

```
type
```

```
TForm1 = class(TForm)
SMS1 : TOxygenSMS;
suiForm1 : TsuiForm;
suiFileTheme1 : TsuiFileTheme;
StBar : TsuiStatusBar;
Panel1 : TPanel;
suiToolBar1 : TsuiToolBar;
DisConnBtn : TsuiButton;
ExitBtn : TsuiButton;
Panel2 : TPanel;
Panel3 : TPanel;
AboutBtn : TsuiButton;
EntryBtn : TsuiButton;
ConnBtn : TsuiButton;
ShowBtn : TsuiButton;
DataSource1 : TDataSource;
OpenPictureDialog1 : TOpenPictureDialog;
AMWaitingGradient1 : TAMWaitingGradient;
ReportBtn : TsuiButton;
DEIBtn : TsuiButton;
AbsenBtn : TsuiButton;
ScrollingCaption1 : TScrollingCaption;
AbsenTimer : THiTimer;
PacketComPort1 : TPacketComPort;
AmEnterAsTab1 : TAmEnterAsTab;
AMDBF : TAMDBF;
Panel4 : TPanel;
suiPanel1 : TsuiPanel;
PortL : TLabel;
ConnTypeL : TLabel;
suiPanel4 : TsuiPanel;
Label4 : TLabel;
Label22 : TLabel;
BattBar : TsuiProgressBar;
SignalBar : TsuiProgressBar;
```

---

PortE	: TSpinEdit;
ConnTCB	: TsuiComboBox;
AboutPanel	: TsuiPanel;
CloseBtn	: TsuiButton;
AbsenPanel	: TsuiPanel;
Image3	: TImage;
suiPanel6	: TsuiPanel;
Image2	: TImage;
ClsAbsenBtn	: TsuiButton;
ReportPanel	: TsuiPanel;
Bevel2	: TBevel;
Label19	: TLabel;
Label20	: TLabel;
Label21	: TLabel;
Label28	: TLabel;
Label29	: TLabel;
Label30	: TLabel;
ClsViewBtn	: TsuiButton;
KirimBtn	: TsuiButton;
suiDBGrid1	: TsuiDBGrid;
DataPanel	: TsuiPanel;
suiPanel2	: TsuiPanel;
Label9	: TLabel;
Label10	: TLabel;
Label11	: TLabel;
Label12	: TLabel;
Label13	: TLabel;
Label14	: TLabel;
Label16	: TLabel;
Label17	: TLabel;
Label18	: TLabel;
Label27	: TLabel;
Label31	: TLabel;
suiPanel3	: TsuiPanel;
Image1	: TImage;
suiDBNavigator1	: TsuiDBNavigator;
suiDBEdit1	: TsuiDBEdit;
suiDBEdit2	: TsuiDBEdit;
suiDBEdit3	: TsuiDBEdit;
suiDBEdit4	: TsuiDBEdit;
suiDBEdit6	: TsuiDBEdit;
suiDBEdit7	: TsuiDBEdit;
FotoBtn	: TsuiButton;
FotoEdit	: TsuiDBEdit;
ActivTagBtn	: TsuiButton;
suiDBEdit9	: TsuiDBEdit;
suiDBEdit10	: TsuiDBEdit;
Label32	: TLabel;

---

```

Label33           : TLabel;
Memo1             : TMemo;
NewRepBtn         : TsuiButton;
SendBar          : TsuiProgressBar;
Label1           : TLabel;
Image4           : TImage;
AmProtector1     : TAmProtector;
suiDBComboBox1   : TsuiDBComboBox;
suiButton2       : TsuiButton;
suiButton3       : TsuiButton;
suiButton1       : TsuiButton;
AbsenTabel      : TAMDBF;
AbsenTabelINDEX  : TSmallintField;
AbsenTabelTANGGAL : TDateField;
AbsenTabelNO_INDUK : TLargeintField;
AbsenTabelNAMA   : TStringField;
AbsenTabelKELAS  : TStringField;
AbsenTabelJENIS  : TStringField;
AbsenTabelENTRY  : TStringField;
AbsenTabelREPORT_SMS : TStringField;
AbsenTabelNO_HP  : TStringField;
AbsenSrc         : TDataSource;
AMDBFINDEX       : TSmallintField;
AMDBFNO_INDUK    : TLargeintField;
AMDBFTAG_FISIK   : TStringField;
AMDBFTAG_ELEC    : TStringField;
AMDBFNAMA        : TStringField;
AMDBFKELAS       : TStringField;
AMDBFJENIS       : TStringField;
AMDBFFOTO        : TStringField;
AMDBFNAMA_SING   : TStringField;
AMDBFNO_HP       : TStringField;
suiImagePanel1   : TsuiImagePanel;
Label3           : TLabel;
Label23          : TLabel;
Label2           : TLabel;
Label24          : TLabel;
Label25          : TLabel;
suiProgressBar1  : TsuiProgressBar;
suiButton4       : TsuiButton;
Label5           : TLabel;
ScrollingCaption2 : TScrollingCaption;
procedure FormCreate(Sender: TObject);
procedure ConnBtnClick(Sender: TObject);
procedure DisConnBtnClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure SMSRepReceived(Index: Integer; Time: TDateTime; Send: String;
    Delivery: integer);

```

---

```

procedure SMSMSGReceived(Index: Integer; Time: TDateTime; Text, Send: String;
    Pict: TBitmap);
procedure SendMsgBtnClick(Sender: TObject);
procedure ConnTCBChange(Sender: TObject);
procedure PortEChange(Sender: TObject);
procedure ConnState(st: boolean);
procedure ExitBtnClick(Sender: TObject);
procedure FotoBtnClick(Sender: TObject);
procedure DataSource1DataChange(Sender: TObject; Field: TField);
procedure FormShow(Sender: TObject);
procedure BUTTON1Click(Sender: TObject);
procedure ActivTagBtnClick(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure ShowBtnClick(Sender: TObject);
procedure EntryBtnClick(Sender: TObject);
procedure CloseBtnClick(Sender: TObject);
procedure AboutBtnClick(Sender: TObject);
procedure ClsViewBtnClick(Sender: TObject);
procedure ReportBtnClick(Sender: TObject);
procedure ScrollingCaption1ProcessCaption(Sender: TObject; Text: String);
procedure ClsAbsenBtnClick(Sender: TObject);
procedure AbsenBtnClick(Sender: TObject);
procedure DElBtnClick(Sender: TObject);
procedure KirimBtnClick(Sender: TObject);
procedure AbsenTimerTimer(Sender: TObject);
procedure Dbf1AfterEdit(DataSet: TDataSet);
procedure ComAfterOpen(Sender: TObject);
procedure PacketComPort1LinkCard(Sender: TObject);
procedure PacketComPort1RxSTR(Sender: TObject; Str: String);
procedure Memo1Change(Sender: TObject);
procedure NewRepBtnClick(Sender: TObject);
procedure suiButton1Click(Sender: TObject);
procedure AbsenTabelFilterRecord(DataSet: TDataSet; var Accept: Boolean);
procedure suiButton3Click(Sender: TObject);
procedure suiButton2Click(Sender: TObject);
procedure suiButton4Click(Sender: TObject);
procedure ScrollingCaption2ProcessCaption(Sender: TObject;
    Text: String);

```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```

AbsenEvent      : Boolean;
TagReceive      : Boolean;
IsFoundTag      : Boolean;
SMSResult       : Boolean;
ProsesAbsenCount : Byte;

```

---

```

FPhoneConnection    : boolean;
SendSmsTimeOut      : Boolean;
FSmsOnSending        : Boolean;
fOnAppActive         : Boolean;
FDateShow            : Boolean;
Dummy               : Boolean;
TimeCount            : Integer;
Counter              : Integer;
FRIDTagCode          : String;
Waktu                : string;
StudentAbsen         : String;
FHpNumber            : String;
FDelivery            : Integer;
waktu2               : TDateTime;
Procedure HideLabel;
Procedure ShowStudent;
Function IsTimeOut(Value: integer): Boolean;
procedure AbsenUpdate;
Function ClearAbsen           : Boolean;
Function FindReport           : Boolean;
Function Reporting            : Boolean;
Function SiswaRecCount        : Boolean;
Function FindStudent           : Boolean;
Function FindStudentName      : Boolean;
Function FindStudentAbsent    : Boolean;
Function FindMorning           : Boolean;
Procedure UpdateReport;
end;

```

```
var
```

```
Form1: TForm1;
```

```
implementation
```

```
uses DateUtils, ComObj, Tanggal, List, Unit3;
```

```
{ $R *.DFM }
```

```
function TForm1.IsTimeOut(Value: integer): Boolean;
```

```
begin
```

```
Result := TimeCount > Value;
```

```
If Result Then TimeCount := 0
```

```
Else Inc(TimeCount);
```

```
end;
```

```
procedure TForm1.ConnState(st: boolean);
```

```
begin
```

---

```

    PortE.enabled := not st;
    ConnTCB.Enabled := not st;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    ConnTCB.ItemIndex := 0;
    ConnState(false);
    AMDBF.Active := False;
    AbsenTabel.Active := False;
    SendSmsTimeOut := True;
    FSmsOnSending := False;
end;

procedure TForm1.ConnBtnClick(Sender: TObject);
begin
    SMS1.Close;
    FPhoneConnection := SMS1.Open;
    if FPhoneConnection then
        StBar.Panels[1].text := 'Phone Type : '+ sms1.PhoneType
    Else ShowMessage('Gagal terkoneksi ke Handphone.!!! ');
end;

procedure TForm1.DisConnBtnClick(Sender: TObject);
begin
    SMS1.Close;
    FPhoneConnection := False;
    StBar.Panels[1].text := 'Phone Type : None';
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
Var Count : Integer;
begin
    ScrollingCaption1.Enabled := False;
    AbsenTimer.Enabled := False;
    PacketComPort1.Connected := False;
    SMS1.Close;
    if AMDBF.Active Then AMDBF.Active := false;
    if AbsenTabel.Active Then AbsenTabel.Active := False;
    For Count:= 1 To 50 Do Begin
        Application.ProcessMessages;
        Sleep(0);
    End;
end;

procedure TForm1.SMSRepReceived(Index: Integer; Time: TDateTime; Send: String;
Delivery: integer);
begin

```

---



```

StBar.Panels[2].text := 'Report Masuk';
Memo1.Lines.Add('Report masuk ');
Memo1.Lines.Add('index := '+ IntToStr(Index));
Memo1.Lines.Add('deliver := '+ IntToStr(Delivery));
Memo1.Lines.Add('sender := '+ Send);
FHPNumber := Send;
FDelivery := Delivery;
AbsenTabel.ForEach(FindReport, True);
FSmsOnSending := true;
Try
  If SMS1.DeleteSMSMessage(Index) then StBar.Panels[2].text := 'Report Dihapus'
  else StBar.Panels[2].text := 'Report tidak dapat Dihapus';
except
  StBar.Panels[2].text := 'Report tidak dapat Dihapus';
end;
FSmsOnSending := False;
end;

function TForm1.FindReport: Boolean;
begin
  If (AbsenTabelNO_HP.Value = FHPNumber) And (AbsenTabelTANGGAL.Value =
DateSet) Then Begin
    AbsenTabel.Edit;
    case FDelivery of
      0: AbsenTabelREPORT_SMS.Value := ' Laporan diterima';
      3: AbsenTabelREPORT_SMS.Value := 'Laporan tidak diterima';
      else AbsenTabelREPORT_SMS.Value := IntToStr(FDelivery)+' Laporan gagal
terkirim';
    end;
    AbsenTabel.Post;
    Result := True;
  End Else Result := False;
end;

procedure TForm1.SMSMSGReceived(Index: Integer; Time: TDateTime; Text, Send:
String; Pict: TBitmap);
begin
  StBar.Panels[2].text := 'Pesan masuk';
  Memo1.Lines.Add('sms masuk ');
  Memo1.Lines.Add('index := '+ IntToStr(Index));
  Memo1.Lines.Add('Text := '+ Text);
  Memo1.Lines.Add('sender := '+ Send);
  FSmsOnSending := true;
  try
    If SMS1.DeleteSMSMessage(Index) then begin
      StBar.Panels[2].text := 'Message deleted';
    end else StBar.Panels[2].text := 'Message NOT deleted';
  end;
end;

```

---

```

    except end;
    FSmsOnSending := false;
end;

procedure TForm1.SendMsgBtnClick(Sender: TObject);
begin
    If SMS1.CheckConnection then begin
        StBar.Panels[2].text := 'Sending message...';
        FSmsOnSending := true;
        SMSResult := sms1.SendSMSMessage(AMDBFNO HP.Value,
            'siswa'+AbsenTabelNAMA.Value+'MASUK JAM '+
            AbsenTabelENTRY.Value,167, True, False, nil);
        End Else SMSResult := False;
        FSmsOnSending := False;
        if SMSResult then StBar.Panels[2].text := 'Message sent' else StBar.Panels[2].text :=
            'Message NOT sent';
    end;

procedure TForm1.ConnTCBChange(Sender: TObject);
begin
    SMS1.ConnectionMode := ConnTCB.ItemIndex;
end;

procedure TForm1.PortEChange(Sender: TObject);
begin
    SMS1.ComNumber := StrToInt(PortE.text);
end;

procedure TForm1.ExitBtnClick(Sender: TObject);
begin
    ScrollingCaption1.Enabled := False;
    AbsenTimer.Enabled := False;
    PacketComPort1.Connected := False;
    SMS1.Close;
    if AMDBF.Active Then AMDBF.Active := false;
    if AbsenTabel.Active Then AbsenTabel.Active :=False;
    Application.Terminate;
end;

procedure TForm1.FotoBtnClick(Sender: TObject);
begin
    If AMDBF.Active and((AMDBF.State = dsInsert) or (AMDBF.State = dsEdit)) Then
    If OpenPictureDialog1.Execute Then Begin
        FotoEdit.Field.AsString := OpenPictureDialog1.FileName;
        Image1.Picture.LoadFromFile(OpenPictureDialog1.FileName);
        Image1.Visible := True;
    End;
end;

```

---

```

end;

procedure TForm1.DataSource1DataChange(Sender: TObject; Field: TField);
begin
    Image1.Visible := FileExists(FotoEdit.Field.AsString);
    if Image1.Visible Then Image1.Picture.LoadFromFile(FotoEdit.Field.AsString);
end;

procedure TForm1.FormShow(Sender: TObject);
begin
    suiPanel1.Align := alLeft;
    suiPanel1.Height := Panel4.Height;
    AboutPanel.Height := Panel4.Height;
    try PacketComPort1.Connected := true;
    except
        showmessage(PacketComPort1.port+' digunakan oleh program lain !!');
    end;
    SendSmsTimeOut := True;
    Button1Click(Self);
    Calendar.SetDate(Now);
End;

procedure TForm1.Button1Click(Sender: TObject);
begin
    AMDBF.Active := not AMDBF.Active;
    AbsenTabel.Active := AMDBF.Active;
end;

procedure TForm1.ActivTagBtnClick(Sender: TObject);
begin
    If AMDBF.Active and PacketComPort1.LinkCard Then Begin
        if (AMDBF.State = dsInsert) or (AMDBF.State = dsEdit) Then
            PacketComPort1.ReadingTagCode;
    End
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    AMDBF.Active := True;
    AbsenTabel.Active := AMDBF.Active;
end;

procedure TForm1.ShowBtnClick(Sender: TObject);
begin
    suiPanel2.Caption := 'DATA PRIBADI SISWA';
    DataPanel.Visible := True;
    DataPanel.BringToFront;
    Button1Click(Self);

```

---

```

    AMDBF.ReadOnly := True;
    ActivTagBtn.Visible := False;
    FotoBtn.Visible := False;
    Button2Click(Self);
end;

procedure TForm1.EntryBtnClick(Sender: TObject);
begin
    suiPanel2.Caption := '          DATA ENTRY SISWA ';
    DataPanel.Visible := True;
    DataPanel.BringToFront;
    Button1Click(Self);
    AMDBF.ReadOnly := False;
    ActivTagBtn.Visible := True;
    FotoBtn.Visible := True;
    Button2Click(Self);
end;

procedure TForm1.CloseBtnClick(Sender: TObject);
begin
    AboutPanel.BringToFront;
    AboutPanel.Visible := False;
end;

procedure TForm1.AboutBtnClick(Sender: TObject);
begin
    AboutPanel.BringToFront;
    AboutPanel.Visible := True;
end;

procedure TForm1.ClsViewBtnClick(Sender: TObject);
begin
    ReportPanel.Visible := False;
end;

Function TForm1.FindMorning : Boolean;
var Wkt1 : TDateTime;
Begin
    If(AbsenTabelTANGGAL.Value = DateSet) Then Begin
        Wkt1 := StrToTimeDef(AbsenTabelENTRY.Value, strToTime('23:59:59'));
        If (CompareTime(Waktu2, Wkt1) = 1) Then Begin
            Label33.Caption := AbsenTabelNAMA.Value;
            Waktu2 := Wkt1;
        End;
        Inc(Counter);
    End;
    Result := False;
End;

```

---

```

Procedure TForm1.UpdateReport;
var Count,RecCount : Integer;
Begin
  Label33.Caption := 'Belum ada';
  Waktu2 := StrToTime('23:59:59');
  Counter := 0;
  AbsenTabel.Active := True;
  AbsenTabel.ForEach(FindMorning,True);
  Count := Counter -1;
  AMDBF.Refresh;
  Counter := 0;
  AMDBF.ForEach(SiswaRecCount,True);
  RecCount := Counter-1;
  Label28.Caption := IntToStr(RecCount);
  Label29.Caption := IntToStr(Count);
  Label30.Caption := IntToStr(RecCount - Count);
End;

procedure TForm1.ReportBtnClick(Sender: TObject);
begin
  ReportPanel.BringToFront;
  ReportPanel.Visible := True;
  AMDBF.Active:=true;
  UpdateReport;
End;

function TForm1.SiswaRecCount: Boolean;
begin
  Result := False;
  inc(Counter);
end;

procedure TForm1.ScrollingCaption1ProcessCaption(Sender: TObject;
  Text: String);
  Var hwnd : THandle;
begin
  suiForm1.BarCaption := Text;
  IF FPhoneConnection then begin
    If Not FSmsOnSending Then begin
      battBar.Position := SMS1.BatteryLevel;
      SignalBar.Position := SMS1.SignalLevel;
    End;
    StBar.Panels[0].text := 'Connected';
    If (AbsenPanel.Visible or ReportPanel.Visible) and
      not Self.Active And Not FDateShow Then Begin
      hwnd := FindWindow(nil, pChar(application.Title));
      Application.ProcessMessages;
    End;
  End;
end;

```

---

```

        PostMessage(hwnd, WM_KEYDOWN, VK_RETURN, 1);
        Application.ProcessMessages;
    end;
end
else Begin
    StBar.Panels[1].text := 'Phone Type : None';
    battBar.Position := 0;
    SignalBar.Position := 0;
    StBar.Panels[0].text := 'Not Connected';

End;
If not SendSmsTimeOut Then Begin
    suiProgressBar1.Position := TimeCount;
    suiProgressBar1.Update;
    StBar.Panels[3].text := 'Time Send :'+ IntToStr(TimeCount);
    inc(TimeCount);
    If (TimeCount = 5) And IsFoundTag Then Begin
        PacketComPort1.SendString(StudentAbsen);
        PacketComPort1.SendApply;
        If TimeCount >= suiProgressBar1.Max Then Begin
            SendSmsTimeOut := True;
        End;
    End
End;
End;
End;

procedure TForm1.ClsAbsenBtnClick(Sender: TObject);
begin
    AbsenPanel.Visible := False;
    AbsenEvent := False;
    EntryBtn.Enabled := True;
    DElBtn.Enabled := True;
    ShowBtn.Enabled := True;
end;

procedure TForm1.AbsenBtnClick(Sender: TObject);
begin
    If PacketComPort1.LinkCard Then Begin
        PacketComPort1.ReadingTagCode;
        EntryBtn.Enabled := False;
        DElBtn.Enabled := False;
        ShowBtn.Enabled := False;
        AbsenPanel.BringToFront;
        AbsenPanel.Visible := True;
        AbsenTimer.Enabled := True;
        ProsesAbsenCount := 0;
        AbsenEvent := True;
    End Else StBar.Panels[2].Text := ' It Lose Connecting To Modul Reader';
end;

```

---

```

end;

procedure TForm1.DEIBtnClick(Sender: TObject);
begin
    suiPanel2.Caption := 'MENGHAPUS SELURUH DATA SISWA';
    DataPanel.Visible := True;
    DataPanel.BringToFront;
    If MessageDlg('Membuat Data Siswa Baru ?...',
        mtConfirmation, [mbYes, mbNo], 0) = mrYes then Begin
        AMDBF.EmptyRecord;
        Image1.Visible := False;
    End;
end;

procedure TForm1.KirimBtnClick(Sender: TObject);
begin
    sms1.SendTimeout := 600;
    If AbsenTabel.Active Then Begin

        Counter := 0;
        If FPhoneConnection then AMDBF.ForEach(Reporting, True)
        Else StBar.Panels[2].text := 'Tidak ada Handphone yang terkoneksi';
    End;
end;

function TForm1.Reporting: Boolean;
var Hit : integer;
begin
    Result := False;
    inc(Counter);
    EntryForm.Edit1.Text := AMDBFNAMA.Value;
    AbsenTabel.ForEach(FindStudent, True);
    If Not Dummy Then Begin
        StBar.Panels[2].text := 'Proses Mengirim Pesan Laporan Ke '+AMDBFNO_HP.Value+' ...';

        StBar.Update;
        FSmsOnSending := true;
        SMSResult := sms1.SendSMSMessage(AMDBFNO_HP.Value,
            'SISWA '+ AMDBFNAMA.Value+ ' Tidak Masuk Hari ini', 167, True, False,
            nil);
        FSmsOnSending := false;
        if SMSResult then StBar.Panels[2].text := 'Pesan ke '+ AMDBFNAMA.Value+ '
Terkirim'
        else StBar.Panels[2].text := 'Pesan ke '+ AMDBFNAMA.Value+ ' Tidak
Terkirim...';
    End;
end;

```

---

```

    StBar.Update;
    SendBar.Max := 40;
    For Hit := 1 To 40 DO Begin
        sleepex(10,false);
        SendBar.Position := Hit;
        SendBar.Update;
        Application.ProcessMessages;
    End;
end;
end;

procedure TForm1.AbsenTimerTimer(Sender: TObject);
begin
    if not AbsenTimer.Enabled or Not AbsenEvent then Exit;
    Case ProsesAbsenCount Of
    0 : If PacketComPort1.CardConnected Then Begin
        PacketComPort1.CardEntryMode;
        HideLabel;
        ProsesAbsenCount:= 1;
        Label23.Caption := 'SILAHKAN ABSEN';
        PacketComPort1.ReadingTagCode;
        End;
    1: If PacketComPort1.CardConnected Then Begin
        If IsTimeOut(3)Then Begin
            PacketComPort1.ReadingTagCode;
            ProsesAbsenCount:= 2;
            End;
        End Else ProsesAbsenCount:= 2;

    2: If TagReceive Then Begin
        IsFoundTag := False;
        TagReceive := False;
        AMDBF.First;
        While not AMDBF.Eof Do Begin
            If (Copy(AMDBFTAG_ELEC.Value,1,12) = Copy(FRIDTagCode,2,12))
//02&03
            Or (Copy(AMDBFTAG_ELEC.Value,1,12) =
Copy(FRIDTagCode,1,12))/02
            Then Begin
                ShowStudent;
                suiProgressBar1.Position := suiProgressBar1.Max;
                break;
            End;
            If Not IsFoundTag Then AMDBF.Next;
        End;
        TimeCount := 0;
        ProsesAbsenCount := 3;
    end;
end;

```

---



```

End Else If PacketComPort1.CardConnected Then Begin
  If TimeCount = 2 Then PacketComPort1.CardEntryMode;
  If IsTimeOut(3)Then Begin
    Label23.Caption := 'SILAHKAN ABSEN';
    Label2.Caption := 'SEKALI LAGI';
    Label2.Visible := True;
    PacketComPort1.ReadingTagCode;
  End;
End;
3: Begin
  If IsFoundTag Then Begin
    Inc(TimeCount);
    If (TimeCount mod 5) = 0 Then Begin
      PacketComPort1.SendString(StudentAbsen);
      PacketComPort1.SendApply;
    End;
    If SendSmsTimeOut Then Begin
      ProsesAbsenCount := 0;
      TimeCount := 0;
    End;
  End Else Begin
    Label2.Visible := True;
    Label23.Caption := 'KARTU INDENTITAS ';
    Label2.Caption := 'ANDA TIDAK DIKENALI';
    If TimeCount >= 1 Then PacketComPort1.SendChar(chr($E4));mk
    If IsTimeOut(5)Then ProsesAbsenCount := 0;
  End;
End;
4: If IsTimeOut(5)Then ProsesAbsenCount := 0;
End;
end;

procedure TForm1.DbflAfterEdit(DataSet: TDataSet);
begin
  Image1.Visible := FileExists(AMDBFFOTO.Value);
end;

procedure TForm1.ComAfterOpen(Sender: TObject);
begin
  AMWaitingGradient1.Enabled := True;
  ScrollingCaption1.Enabled := True;
  Button2Click(Self);
end;

procedure TForm1.PacketComPort1LinkCard(Sender: TObject);
begin
  AMWaitingGradient1.Enabled := False;
  AMWaitingGradient1.Visible := False;

```

---

```

    StBar.Panels[2].text := 'Tag Reader Detected...';
    PacketComPort1.LinkCard := True;
    Button2Click(Self);
end;

function HexToDec(Str: string): Integer;
var
    i, M: Integer;
begin
    Result:=0;
    M:=1;
    Str:=AnsiUpperCase(Str);
    for i:=Length(Str) downto 1 do
        begin
            case Str[i] of
                '1'..'9': Result:=Result+(Ord(Str[i])-Ord('0'))*M;
                'A'..'F': Result:=Result+(Ord(Str[i])-Ord('A')+10)*M;
            end;
            M:= M shl 4;
        end;
    end;
end;

procedure TForm1.PacketComPort1RxSTR(Sender: TObject; Str: String);
Var COUNT : integer;
    S,terima : string;
begin
    s := '';
    For Count := 1 To Length(str) do begin
        s:= s+ intToStr(byte(str[count]))+' ';
    End;
    FRIDTagCode := Str;
    Terima:=(Copy(FRIDTagCode,4,8));
    StBar.Panels[2].text := Str + s + ' '+IntToStr(HexToDec(Terima));
    If AbsenEvent Then TagReceive := True Else
        if (AMDBF.State = dsInsert) or (AMDBF.State = dsEdit) Then Begin
            AMDBFTAG_ELEC.Value := Str;
            AMDBFTAG_FISIK.Value:=
IntToStr(HexToDec((Copy(FRIDTagCode,4,8))));
        End;
    end;

procedure TForm1.HideLabel;
begin
    Label3.Visible := False;
    Label24.Visible := False;
    Label25.Visible := False;
    suiProgressBar1.Visible := False;
    suiProgressBar1.Position := 0;

```

---

```

Label2.Visible := False;
Image2.Visible := False;
TimeCount := 0;
Label3.Update;
Label24.Update;
Label25.Update;
suiProgressBar1.Update;
Label2.Update;
Image2.Update;

end;

Function TForm1.FindStudent : Boolean;
Begin
  Dummy := (AbsenTabelNAMA.Value = AMDBFNAMA.Value) And
  (AbsenTabelTANGGAL.Value = DateSet);
  If Dummy Then Begin
    AbsenTabel.Edit;
    AbsenTabelENTRY.Value := Waktu;
    AbsenTabelREPORT_SMS.Value := 'Mengirim Laporan';
    AbsenTabel.Post;
  End;
  INC (Counter);
  Result := Dummy;
End;

procedure TForm1.ShowStudent;
begin
  IsFoundTag := True;
  Waktu := TimeToStr(Time);
  suiProgressBar1.Position := 0;
  suiProgressBar1.Visible := True;
  application.ProcessMessages;
  StudentAbsen := AMDBFNAMA_SING.Value;
  If FileExists(AMDBFFOTO.Value) Then Begin
    Image2.Picture.LoadFromFile(AMDBFFOTO.Value);
    Image2.Visible := True;
  End;
  Label23.Caption := AMDBFNAMA.Value;
  Label24.Caption := 'JAM MASUK ' + TimeToStr(Time);
  TimeCount := 0;
  Counter := 0;
  AbsenTabel.ForEach(FindStudent, False);
  If Not Dummy Then Begin
    AbsenTabel.Append;
    AbsenTabelINDEX.Value := Counter;
    AbsenTabelNAMA.Value := AMDBFNAMA.Value;
    AbsenTabelKELAS.Value := AMDBFKEELAS.Value;
  End;
end;

```

---

```

    AbsenTabelTANGGAL.Value:= DateSet;
    AbsenTabelNO_INDUK.Value := AMDBFNO_INDUK.Value;
    AbsenTabelNO_HP.Value := AMDBFNO_HP.Value;
    AbsenTabelENTRY.Value := Waktu;
    AbsenTabelREPORT_SMS.Value := 'Mengirim Laporan',
    AbsenTabel.Post;
    UpdateReport;
End;
SendSmsTimeOut := False;
Label3.Visible := True;
Label24.Visible := True;
Label25.Visible := True;
Label2.Visible := False;
PacketComPort1.SendString(StudentAbsen); //LCDWRITERAM
PacketComPort1.SendApply; //E2
FSmsOnSending := True;
If FPhoneConnection then
    SMSResult := sms1.SendSMSMessage(AMDBFNO_HP.Value,
        'SISWA '+ AMDBFNAMA.Value+' Masuk Sekolah Jam '+Waktu,167,
True, False, nil)
    Else SMSResult := False;
    FSmsOnSending := False;
    if SMSResult then begin
        StBar.Panels[2].text := 'Pesan Dikirim';
        SendSmsTimeOut := True;
    End
    else Begin
        StBar.Panels[2].text := 'Pesan Tidak Terkirim';
        AbsenTabel.Edit;
        AbsenTabelREPORT_SMS.Value := 'Laporan Gagal';
        AbsenTabel.Post;
        SendSmsTimeOut := True;
    End;
end;

procedure TForm1.Memo1Change(Sender: TObject);
begin
    if memo1.Lines.Count > 100 Then Memo1.Clear;
end;

procedure TForm1.NewRepBtnClick(Sender: TObject);
begin
    If MessageDlg('Membuat Laporan Siswa Baru ?...',
        mtConfirmation, [mbYes, mbNo], 0) = mrYes then Begin
        AbsenTabel.ForEach(ClearAbsen,False);
        Label28.Caption := '';
        Label30.Caption := '';
        Label29.Caption := '';
    End;
end;

```

---

```
        Label33.Caption := '';
    End;
end;

function TForm1.ClearAbsen: Boolean;
begin
    Result := False;
    If AbsenTabelTANGGAL.Value = DateSet Then AbsenTabel.Delete;
end;

procedure TForm1.suiButton1Click(Sender: TObject);
begin
    Calendar.Position := poMainFormCenter;
    FDateShow      := True;
    If Calendar.showModal= IdOk Then AbsenUpdate;
    UpdateReport;
    FDateShow      := False;
End;

procedure TForm1.AbsenUpdate;
begin
    AbsenTabel.FilterUpdate;
    suiDBGrid1.Update;
end;

procedure TForm1.AbsenTabelFilterRecord(DataSet: TDataSet;
    var Accept: Boolean);
begin
    Accept := AbsenTabelTANGGAL.Value = DateSet;
end;

procedure TForm1.suiButton3Click(Sender: TObject);
begin
    ListForm.Judul.Caption := 'DAFTAR ABSENSI SISWA HARI '+
    Calendar.StringDate;
    ListForm.preview;
end;

procedure TForm1.suiButton2Click(Sender: TObject);
begin
    ListForm.Judul.Caption := 'DAFTAR ABSENSI SISWA HARI '+
    Calendar.StringDate;
    ListForm.printed;
end;

Function TForm1.FindStudentAbsent : Boolean;
var S1,S2 : String;
    s : string;
```

---

```

    i : Integer;
Begin
    s1 := "";
    S2 := "";
    s := AbsenTabelNAMA.Value;
    for i := 1 to Length(s) do s1 := S1 + Uppcase(s[i]);
    s := EntryForm.Edit1.Text;
    for i := 1 to Length(s) do s2 := S2 + Uppcase(s[i]);
    Dummy := s1 = s2;
    Result := Dummy;
End;
procedure TForm1.suiButton4Click(Sender: TObject);

begin
    If EntryForm.ShowModal = MrOk then Begin
        AbsenTabel.ForEach(FindStudentAbsent,False);
        If Dummy Then
            Showmessage(AbsenTabelNAMA.Value+ ' Hari Ini Masuk Jam '+
AbsenTabelENTRY.Value )
        Else Begin
            AMDBF.ForEach(FindStudentName,False);
            If Dummy Then Showmessage(EntryForm.Edit1.Text+ ' Tidak Masuk Hari Ini')
            Else Showmessage('Maaf Siswa Dengan Nama '+EntryForm.Edit1.Text+ ' Tidak
Terdaftar');
        End;
        End;
    end;
Function TForm1.FindStudentName : Boolean;
var S1,S2 : String;
    s : string;
    i : Integer;
Begin
    s1 := "";
    S2 := "";
    s := AMDBFNAMA.Value;
    for i := 1 to Length(s) do s1 := S1 + Uppcase(s[i]);
    s := EntryForm.Edit1.Text;
    for i := 1 to Length(s) do s2 := S2 + Uppcase(s[i]);
    Dummy := s1 = s2;
    Result := Dummy;
End;
procedure TForm1.ScrollingCaption2ProcessCaption(Sender: TObject;
    Text: String);
begin
    label5.Caption:=text;
end;

end.

```

---

```

unit List;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, ExtCtrls, quickrpt, DB, DBTables, Qrctrls, jpeg;

type
  TListForm = class(TForm)
    QuickRep      : TQuickRep;
    DetailBand1   : TQRBand;
    ColumnHeaderBand1: TQRBand;
    PageFooterBand1 : TQRBand;
    TitleBand1    : TQRBand;
    QRDBText1     : TQRDBText;
    QRLabel1      : TQRLabel;
    QRDBText2     : TQRDBText;
    QRLabel2      : TQRLabel;
    QRDBText3     : TQRDBText;
    QRLabel3      : TQRLabel;
    QRDBText4     : TQRDBText;
    QRLabel4      : TQRLabel;
    QRLabel5      : TQRLabel;
    QRDBText5     : TQRDBText;
    Judul         : TQRLabel;
    QRSysData1    : TQRSysData;
    QRSysData2    : TQRSysData;
    QRLabel9      : TQRLabel;
    QRDBText6     : TQRDBText;
    QRLabel7      : TQRLabel;
    QRLabel8      : TQRLabel;
    QRLabel10     : TQRLabel;
    QRLabel11     : TQRLabel;
  private
    { Private declarations }
  public
    { Public declarations }
    Procedure Preview;
    Procedure Printed;
  end;

var
  ListForm: TListForm;

implementation
uses Main;
{$R *.dfm}

```

---

```
{ TListForm }
```

```
procedure TListForm.Preview;
```

```
begin
```

```
    QRLabel7.Caption := 'TOTAL SISWA YANG TERDAFTAR =  
                        '+Form1.Label28.Caption;
```

```
    QRLabel8.Caption := 'TOTAL SISWA YANG MASUK      =  
                        '+Form1.Label29.Caption;
```

```
    QRLabel10.Caption := 'TOTAL SISWA YANG TIDAK MASUK =  
                        '+Form1.Label30.Caption;
```

```
    QRLabel11.Caption := 'SISWA YANG MASUK TERPAGI   =  
                        '+Form1.Label33.Caption;
```

```
    QuickRep.Preview;
```

```
end;
```

```
procedure TListForm.Printed;
```

```
begin
```

```
    QRLabel7.Caption := 'TOTAL SISWA YANG TERDAFTAR =  
                        '+Form1.Label28.Caption;
```

```
    QRLabel8.Caption := 'TOTAL SISWA YANG MASUK      =  
                        '+Form1.Label29.Caption;
```

```
    QRLabel10.Caption := 'TOTAL SISWA YANG TIDAK MASUK =  
                        '+Form1.Label30.Caption;
```

```
    QRLabel11.Caption := 'SISWA YANG MASUK TERPAGI   =  
                        '+Form1.Label33.Caption;
```

```
    QuickRep.Print;
```

```
end;
```

```
end.
```

```
unit Tanggal;
```

```
interface
```

```
uses
```

```
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
    Dialogs, ExtCtrls, SUIButton, ToolWin, ComCtrls, SUIToolBar, Grids,  
    Calendar, SUIForm;
```

```
Const
```

```
    DayName : Array [0..6] Of String
```

```
=( 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu' );
```

```
    MonthName : Array [1..12] Of String =( 'January', 'Febuary', 'Maret', 'April', 'Mei', 'Juni',  
    'July', 'Agustus', 'September', 'Oktober', 'November', 'Desember' );
```

```
type
```

```
    TForm2 = class(TForm)
```

```
        suiForm1 : TsuiForm;
```



```

suiToolBar1 : TsuiToolBar;
suiToolBar2 : TsuiToolBar;
suiButton1   : TsuiButton;
suiButton2   : TsuiButton;
suiButton3   : TsuiButton;
suiButton4   : TsuiButton;
suiButton5   : TsuiButton;
suiButton6   : TsuiButton;
Panel1       : TPanel;
Panel2       : TPanel;
Calendar1    : TCalendar;
procedure suiButton2Click(Sender: TObject);
procedure suiButton1Click(Sender: TObject);
procedure suiButton6Click(Sender: TObject);
procedure suiButton5Click(Sender: TObject);
procedure suiButton4Click(Sender: TObject);
procedure suiButton3Click(Sender: TObject);
procedure FormShow(Sender: TObject);
private
  { Private declarations }
public
  Procedure SaveDate;
  Procedure SetDate(Value : TDateTime);
  Function StringDate : String;
  { Public declarations }
end;

var
  Calendar: TForm2;
  DateSet : TDateTime;

implementation
{$R *.dfm}
Function TForm2.StringDate : String;
Begin
  Result := DayName[Calendar1.StartOfWeek]+' '+
    IntToStr(Calendar1.Day)+' '+
    MonthName[Calendar1.Month] +' '+
    IntToStr(Calendar1.Year);

End;
procedure TForm2.suiButton2Click(Sender: TObject);
begin
  ModalResult := idCancel;
end;

procedure TForm2.suiButton1Click(Sender: TObject);
begin

```

---

```
    DateSet := EncodeDate(Calendar1.Year, Calendar1.Month, Calendar1.Day);
    ModalResult := idOk;
end;

procedure TForm2.suiButton6Click(Sender: TObject);
begin
    Calendar1.Year := Calendar1.Year + 1;
    Panel1.Caption := MonthName[Calendar1.Month] + ' ' + IntToStr(Calendar1.Year);
end;

procedure TForm2.suiButton5Click(Sender: TObject);
begin
    Calendar1.Month := Calendar1.Month + 1;
    Panel1.Caption := MonthName[Calendar1.Month] + ' ' + IntToStr(Calendar1.Year);
end;

procedure TForm2.suiButton4Click(Sender: TObject);
begin
    Calendar1.Month := Calendar1.Month - 1;
    Panel1.Caption := MonthName[Calendar1.Month] + ' ' + IntToStr(Calendar1.Year);
end;

procedure TForm2.suiButton3Click(Sender: TObject);
begin
    Calendar1.Year := Calendar1.Year - 1;
    Panel1.Caption := MonthName[Calendar1.Month] + ' ' + IntToStr(Calendar1.Year);
end;

procedure TForm2.FormShow(Sender: TObject);
begin
    Panel1.Caption := MonthName[Calendar1.Month] + ' ' + IntToStr(Calendar1.Year);
end;

procedure TForm2.SaveDate;
begin
    DateSet := EncodeDate(Calendar1.Year, Calendar1.Month, Calendar1.Day);
end;

procedure TForm2.SetDate(Value : TDateTime);
begin
    Calendar1.CalendarDate := Value;
    DateSet := EncodeDate(Calendar1.Year, Calendar1.Month, Calendar1.Day);
end;

end.
```

---

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, SUIButton, ExtCtrls, SUIForm;

type
  TEntryForm = class(TForm)
    suiForm1: TSuiForm;
    suiButton1: TSuiButton;
    suiButton2: TSuiButton;
    Label1: TLabel;
    Edit1: TEdit;
    Bevel1: TBevel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  EntryForm: TEntryForm;

implementation
uses main;
{$R *.dfm}

end.
```

---

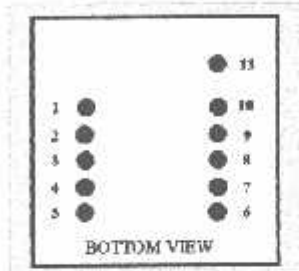
# ID SERIES DATASHEET

Feb 10 , 2004

## ID-2 / ID-12 / ID-20

The ID2, ID12 and ID20 are similar to the ID0, ID10 and ID15 vK(ii) series devices, but they have extra pins which allow Magnetic Emulation output to be included in the functionality. The ID-12 and ID-20 come with internal antennas, and have read ranges of 12+ cm and 16+ cm, respectively. With an external antenna, the ID-2 can deliver read ranges of up to 25 cm. All three readers support ASCII, Wiegand26 and Magnetic ABA Track2 data formats.

### ID2 / ID12 / ID20 PIN-OUT



- 1. GND
- 2. RES (Reset Bar)
- 3. ANT (Antenna)
- 4. ANT (Antenna)
- 5. CP
- 6. Future
- 7. +/- (Format Selector)
- 8. D1 (Data Pin 1)
- 9. D0 (Data Pin 0)
- 10. LED (LED / Beeper)
- 11. +5V



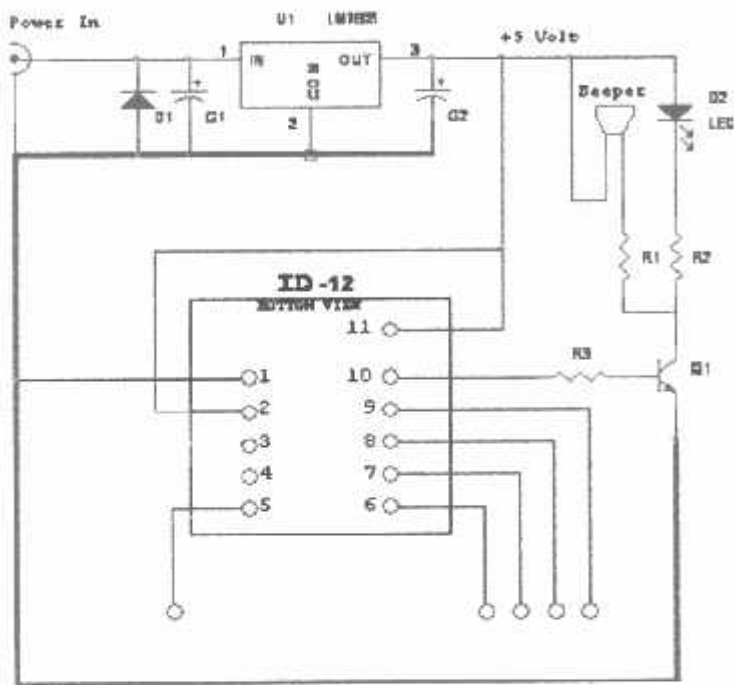
## Operational and Physical Characteristics

Parameters	ID-2	ID-12	ID-20
Read Range	N/A (no internal antenna)	12+ cm	16+ cm
Dimensions	21 mm x 19 mm x 6 mm	26 mm x 25 mm x 7 mm	40 mm x 40 mm x 9 mm
Frequency	125 kHz	125 kHz	125 kHz
Card Format	EM 4001 or compatible	EM 4001 or compatible	EM 4001 or compatible
Encoding	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64
Power Requirement	5 VDC @ 13mA nominal	5 VDC @ 30mA nominal	5 VDC @ 85mA nominal
I/O Output Current	+/-200mA PK	-	-
Voltage Supply Range	+4.6V through +5.4V	+4.6V through +5.4V	+4.6V through +5.4V

## Pin Description & Output Data Formats

Pin No.	Description	ASCII	Magnet Emulation	Wiegand26
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V	GND 0V
Pin 2	Strap to +5V	Reset Bar	Reset Bar	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna	Antenna
Pin 5	Card Present	No function	Card Present	No function
Pin 6	Future	Future	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND	Strap to Pin 10	Strap to +5V
Pin 8	Data 1	CMOS	Clock	One Output
Pin 9	Data 0	TTL Data (inverted)	Data	Zero Output
Pin 10	3.1 kHz Logic	Beeper / LED	Beeper / LED	Beeper / LED
Pin 11	DC Voltage Supply	+5V	+5V	+5V

Circuit Diagram for the ID-12



COMPONENT LIST

- R1 = 100R
- R2 = 1K
- R3 = 1K
- C1 = 100uF 16V
- C2 = 100uF 10V
- Beeper = 2.7-3.5KHz 100R
- D1 = 1N4001
- D2 = GREEN LED
- U1 = LM7805
- Q1 = UTC8050 (NPN)
- ID2 = ID Innovations ID2

\* Please Note the ID2 has an internal tuning capacitor of 1.5nF and this makes the total tuning capacity = 2.5nF

The 3.1KHz Beeper Logic is centered for most Beepers in range 2.7-3.5KHz

DATA FORMATS

Output Data Structure – ASCII

STX (02h)	DATA (10 ASCII)	CHECK SUM (2 ASCII)	CR	LF	ETX (03h)
-----------	-----------------	---------------------	----	----	-----------

The 1byte (2 ASCII characters) Check sum is the "Exclusive OR" of the 5 hex bytes (10 ASCII) Data characters.]

Output Data Structure – Wiegand26

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P	E	E	E	E	E	E	E	E	E	E	E	E	O	O	O	O	O	O	O	O	O	O	O	O	P
Even parity (E)													Odd parity (O)												

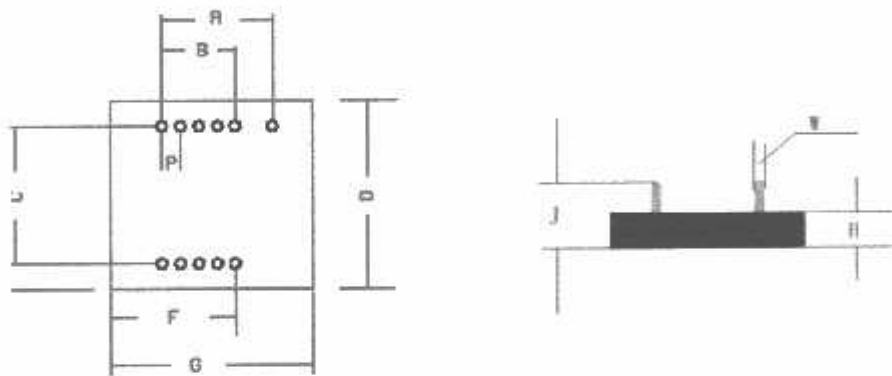
= Parity start bit and stop bit

Output Data Magnetic ABA Track2

10 Leading Zeros	SS	Data	ES	LCR	10 Ending Zeros
------------------	----	------	----	-----	-----------------

SS is the Start Character of 11010, ES is the end character of 11111, LRC is the Longitudinal Redundancy Check.]

Dimensions (Top View) (mm)



ID-0/ID-2			ID-10/ID-12			ID-15/ID-20		
Nom.	Min.	Max.	Nom.	Min.	Max.	Nom.	Min.	Max.
12.0	11.6	12.4	12.0	11.6	12.4	12.0	11.6	12.4
8.0	7.6	8.4	8.0	7.6	8.4	8.0	7.6	8.4
15.0	14.6	15.4	15.0	14.6	15.4	15.0	14.6	15.4
20.5	20.0	21.5	25.3	24.9	25.9	40.3	40.0	41.0
18.5	18.0	19.2	20.3	19.8	20.9	27.8	27.5	28.5
14.0	13.0	14.8	16.3	15.8	16.9	22.2	21.9	23.1
22.0	21.6	22.4	26.4	26.1	27.1	38.5	38.2	39.2
2.0	1.8	2.2	2.0	1.8	2.2	2.0	1.8	2.2
5.92	5.85	6.6	6.0	5.8	6.6	6.8	6.7	7.0
9.85	9.0	10.5	9.9	9.40	10.5	9.85	9.4	10.6
0.66	0.62	0.67	0.66	0.62	0.67	0.66	0.62	0.67

Note – measurements do not include any burring of edges.  
NOTICE - Innovated Devices reserve the right to change these specifications without prior notice.

## Features

- Compatible with MCS-51™ Products
- Features of In-System Reprogrammable Downloadable Flash Memory
- SPI Serial Interface for Program Downloading
- Endurance: 1,000 Write/Erase Cycles
- Features EEPROM
- Endurance: 100,000 Write/Erase Cycles
- 5V Operating Range
- Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 8-bit Internal RAM
- Programmable I/O Lines
- 16-bit Timer/Counters
- Interrupt Sources
- Programmable UART Serial Channel
- Serial Interface
- Lower Idle and Power-down Modes
- Fast Recovery From Power-down
- Programmable Watchdog Timer
- Data Pointer
- Watch-off Flag

## Description

AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of downloadable Flash programmable and erasable read only memory and 512 bytes of EEPROM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 pinout. The on-chip downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a traditional nonvolatile memory programmer. By combining a versatile 8-bit CPU with downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 512 bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode allows the CPU to be shut down while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but disables the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from. Lock Bit 2 has been activated.



## 8-bit Microcontroller with 8K Bytes Flash

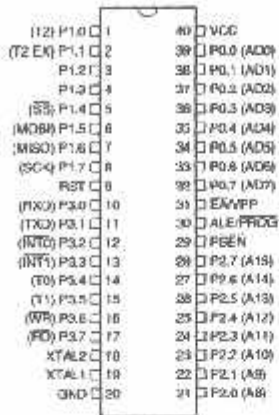
**AT89S8252**

Rev. 0401E -02/00

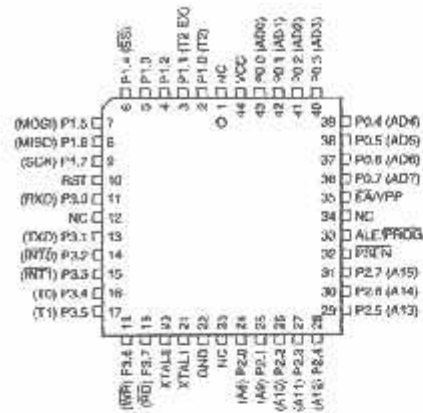


## Configurations

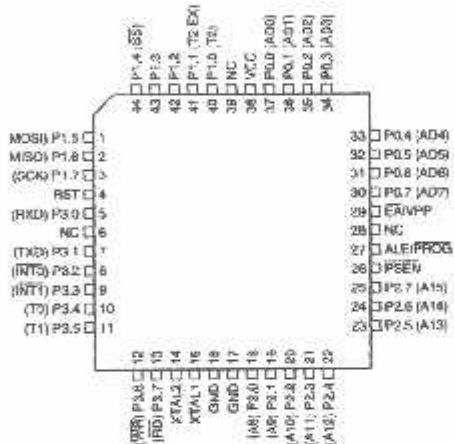
PDIP



PLCC



PQFP/TQFP



## Description

voltage.

is an 8-bit open drain bi-directional I/O port. As an output, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

can also be configured to be the multiplexed low-address/data bus during accesses to external

program and data memory. In this mode, P0 has internal pullups.

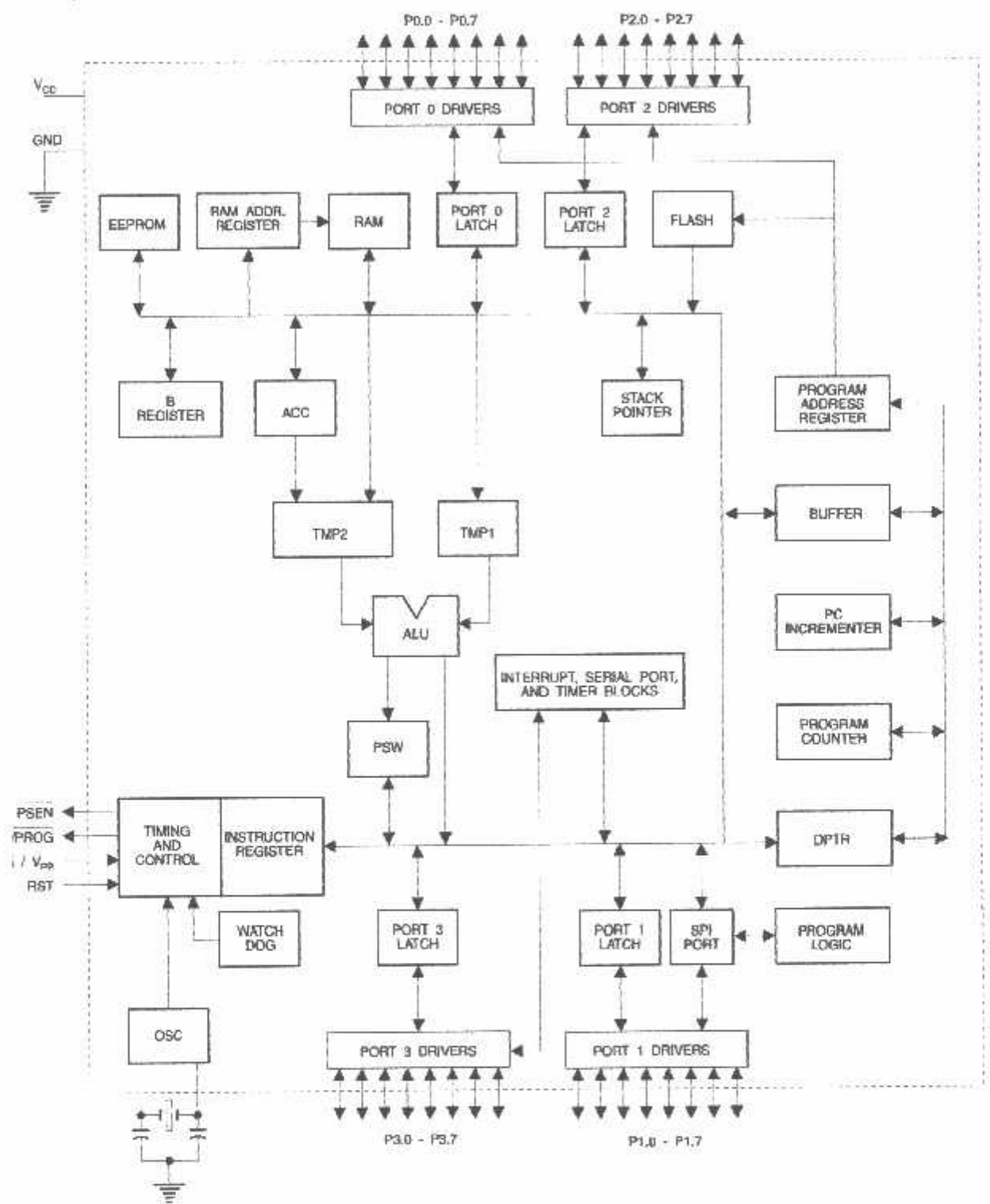
Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

### Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the internal pullups.



k Diagram



Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

### Description

Ports P1.4, P1.5, P1.6, and P1.7 can be configured SPI slave port select, data input/output and shift clock/output pins as shown in the following table.

Pin	Alternate Functions
P1.2	T2 (external count input to Timer/Counter 2), clock-out
P1.3	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	SS (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 2 also receives the low-order address bytes during Flash programming and verification.

Port 2 is an 8-bit bi-directional I/O port with internal pullups. When 2 output buffers can sink/source four TTL inputs. When 2 outputs are written to Port 2 pins, they are pulled high by internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-up emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 outputs the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3 is an 8-bit bi-directional I/O port with internal pullups. When 3 output buffers can sink/source four TTL inputs. When 3 outputs are written to Port 3 pins, they are pulled high by internal pullups and can be used as inputs. As inputs,

Port 3 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

### RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

### ALE/PROG

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

### PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

### EA/VPP

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external pro-

memory locations starting at 0000H up to FFFFH. However, that if lock bit 1 is programmed,  $\overline{EA}$  will be latched on reset.

It should be strapped to  $V_{CC}$  for internal program execution. This pin also receives the 12-volt programming voltage ( $V_{pp}$ ) during Flash programming when 12V programming is selected.

- XTAL1**  
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.
- XTAL2**  
Output from the inverting oscillator amplifier.

AT89S8252 SFR Map and Reset Values

								0FFH
B 00000000								0F7H
								0EFH
ACC 00000000								0E7H
								0DFH
PSW 00000000					SPCR 000001XX			0D7H
T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
								0C7H
IP XX000000								0BFH
P3 11111111								0B7H
IE 0X000000		SPSR 00XXXXXX						0AFH
P2 11111111								0A7H
SCON 00000000	SBUF XXXXXXXX							9FH
P1 11111111						WMCON 00000010		97H
TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H





## Special Function Registers

of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

at not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Accesses to these addresses will in general return data, and write accesses will have an indeterminate

software should not write 1s to these unlisted

locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

**Timer 2 Registers** Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

Table 2. T2CON—Timer/Counter 2 Control Register

Address = 0C8H

Reset Value = 0000 0000B

Writable

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
7	6	5	4	3	2	1	0

Bit	Function
7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
5	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
4	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
3	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
1	Timer or counter select for Timer 2. C/T2 = 0 for timer function, C/T2 = 1 for external event counter (falling edge triggered).
0	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

**log and Memory Control Register** The WMCON register contains control bits for the Watchdog Timer (in Table 3). The EEMEN and EEMWE bits are used

to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

**WMCON—Watchdog and Memory Control Register**

IN Address = 96H

Reset Value = 0000 0010B

PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
7	6	5	4	3	2	1	0

Bit	Function
7:5	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
4	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
3	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
2	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1.
1:0	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.
0	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.

**Registers** Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data during serial data transfer sets the Write Collision Flag, in the SPSR register. The SPDR is double buffered and the values in SPDR are not changed by

**Interrupt Registers** The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In the IPR register, the individual interrupt priority for the SPI is in the IPR register. Two priorities can be set for each of the interrupt sources in the IP register.

**Dual Data Pointer Registers** To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

**Power Off Flag** The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.





### SPCR—SPI Control Register

Address = D5H

Reset Value = 0000 01XXB

SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
7	6	5	4	3	2	1	0

#### Function

**SPI Interrupt Enable.** This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.

**SPI Enable.** SPI = 1 enables the SPI channel and connects  $\overline{SS}$ , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.

**Data Order.** DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.

**Master/Slave Select.** MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.

**Clock Polarity.** When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.

**Clock Phase.** The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.

**SPI Clock Rate Select.** These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency,  $F_{osc}$ , is as follows:

SPR1SPR0 SCK =  $F_{osc}$  divided by

0	0	4
0	1	16
1	0	64
1	1	128

### SPSR – SPI Status Register

Address = AAH

Reset Value = 00XX XXXXB

SPIF	WCOL	–	–	–	–	–	–
7	6	5	4	3	2	1	0

#### Function

**SPI Interrupt Flag.** When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.

**Write Collision Flag.** The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

### SPDR – SPI Data Register

Address = 86H

Reset Value = unchanged

SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
7	6	5	4	3	2	1	0

**AT89S8252**



### Memory – EEPROM and RAM

89S8252 implements 2K bytes of on-chip EEPROM storage and 256 bytes of RAM. The upper 128 of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the addresses as the SFR space but are physically separate SFR space.

an instruction accesses an internal location above 0x7FH, the address mode used in the instruction is whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOVX, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses a byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOVX, @R0
```

Stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

On-chip EEPROM data memory is selected by setting the EEN bit in the WMCON register at SFR address 96H. The EEPROM address range is from 000H to 0FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EWE bit in the WMCON register needs to be set to enable any byte location in the EEPROM can be written. Software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the programming mode are self-timed and typically take 4 to 8 ms. The progress of EEPROM write can be monitored using the RDY/BSY bit (read-only) in SFR WMCON. BSY = 0 means programming is still in progress and BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

During EEPROM programming, an attempted read from the EEPROM will fetch the byte being written (MSB complemented). Once the write cycle is complete, the data are valid at all bit locations.

### Programmable Watchdog Timer

Programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1, and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available periods are shown in the following table and the

actual timer periods (at V<sub>CC</sub> = 5V) are within ±30% of the nominal.

The WDT is disabled by Power-on Reset and during Power-down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDTRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

### Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

### Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which

### Timer 2 Operating Modes

+ TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

The diagram illustrates the internal structure of the Timer 2 module. It starts with an oscillator (OSC) feeding into a prescaler (+12). The output of the prescaler is split: one path goes through a switch controlled by  $C/T2 = 0$  to the TH2 register, and another path goes through a switch controlled by  $C/T2 = 1$  to the EXEN2 switch. The EX PIN is connected to a transition detector, which then feeds into the EXEN2 switch. The output of the EXEN2 switch goes to the EXF2 register. The TH2 register is connected to the TL2 register, which has an overflow output. Both TH2 and TL2 are connected to RCAP2H and RCAP2L registers via comparators. The output of the RCAP2H and RCAP2L registers is connected to the EXF2 register. The EXF2 register has an output to the TIMER 2 INTERRUPT. The TR2 control signal is connected to the EXEN2 switch and the RCAP2H and RCAP2L registers.

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.



reload (Up or Down Counter)

2 can be programmed to count up or down when red in its 16 bit auto-reload mode. This feature is l by the DCEN (Down Counter Enable) bit located in 3 T2MOD (see Table 9). Upon reset, the DCEN bit o 0 so that timer 2 will default to count up. When s set, Timer 2 can count up or down, depending on ie of the T2EX pin.

2 shows Timer 2 automatically counting up when = 0. In this mode, two options are selected by bit in T2CON. If EXEN2 = 0, Timer 2 counts up to 4 and then sets the TF2 bit upon overflow. The r also causes the timer registers to be reloaded with it value in RCAP2H and RCAP2L. The values in H and RCAP2L are preset by software. If EXEN2 = bit reload can be triggered either by an overflow or

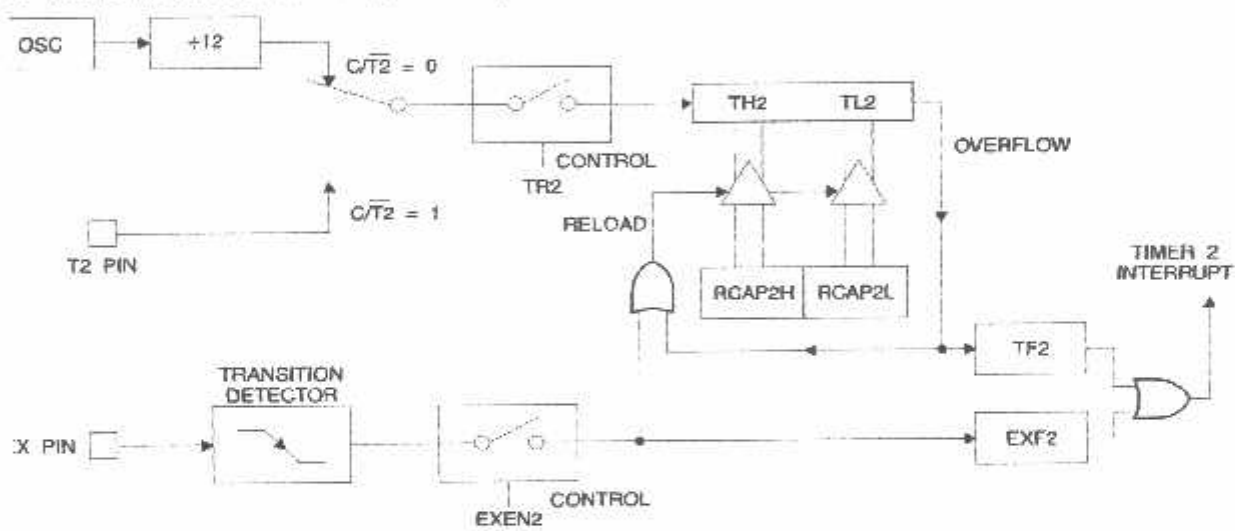
by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

2. Timer 2 in Auto Reload Mode (DCEN = 0)

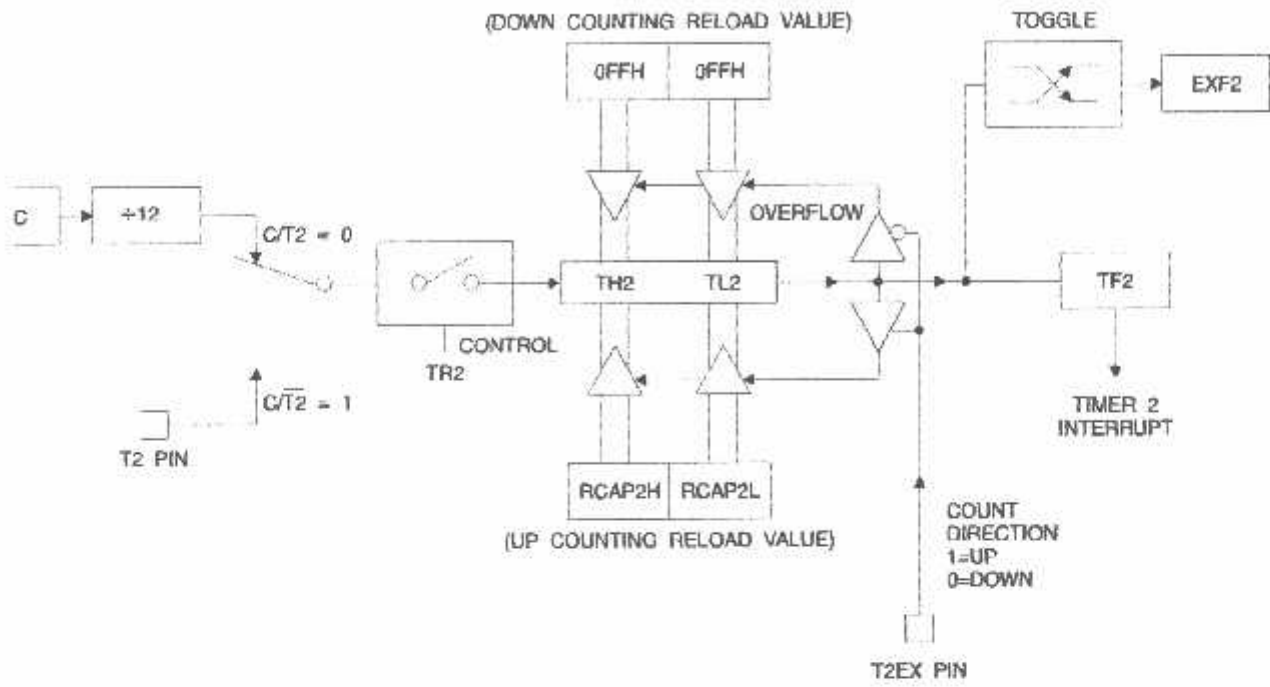


T2MOD – Timer 2 Mode Control Register							
Address = 0C9H						Reset Value = XXXX XX00B	
Addressable							
–	–	–	–	–	–	T2OE	DCEN
7	6	5	4	3	2	1	0

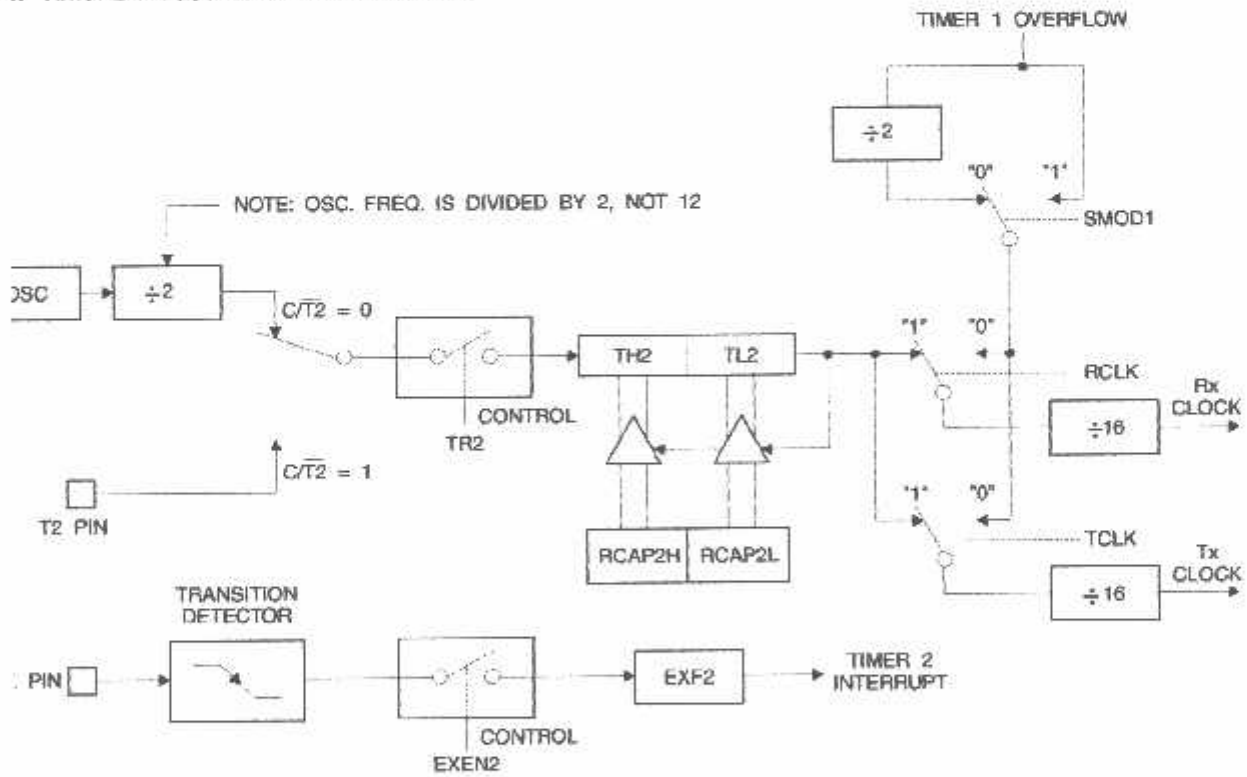
I	Function
	Not implemented, reserved for future use.
	Timer 2 Output Enable bit.
	When set, this bit allows Timer 2 to be configured as an up/down counter.



### 3. Timer 2 Auto Reload Mode (DCEN = 1)



### 1. Timer 2 in Baud Rate Generator Mode



## Rate Generator

Timer 2 is selected as the baud rate generator by setting RCLK and/or TCLK in T2CON (Table 2). Note that the rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

Baud rate generator mode is similar to the auto-reload mode in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

Baud rates in Modes 1 and 3 are determined by Timer 2 overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

Timer 2 can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different from counter when it is used as a baud rate generator. Normally, counter, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\text{Modes 1 and 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (RCAP2H, RCAP2L)]}$$

(RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

When used as a baud rate generator is shown in Figure 4. This formula is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload of RCAP2H, RCAP2L to (TH2, TL2). Thus when Timer

2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

## Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

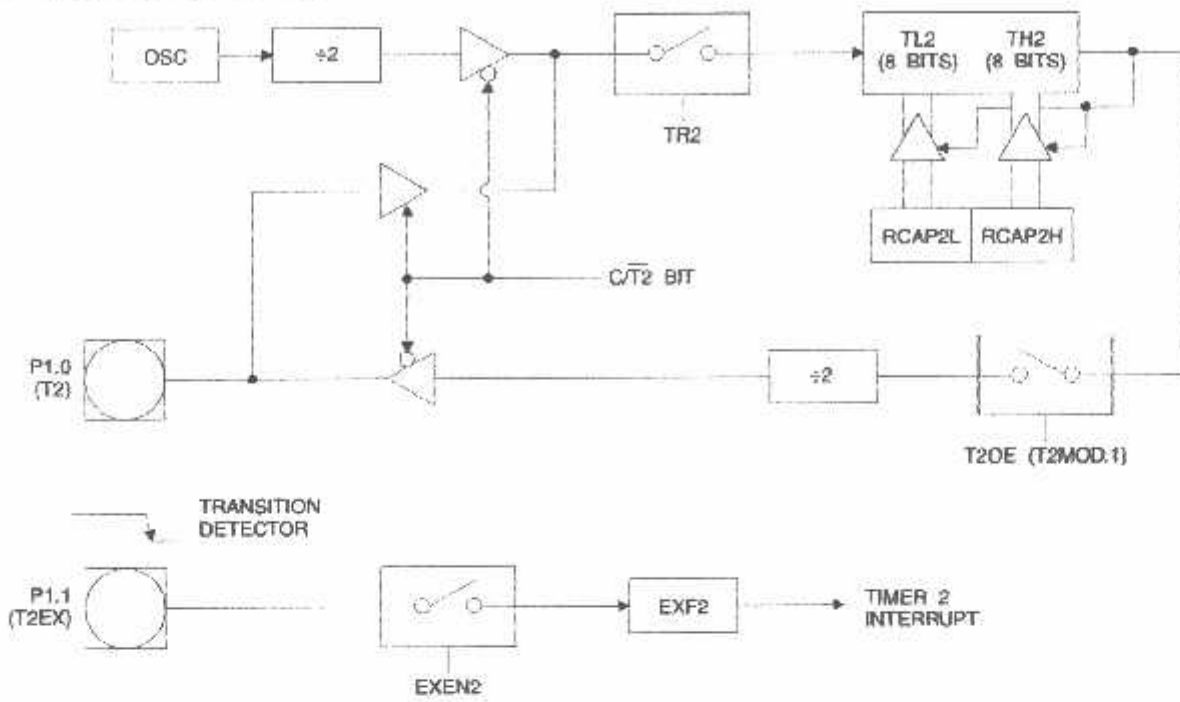
To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

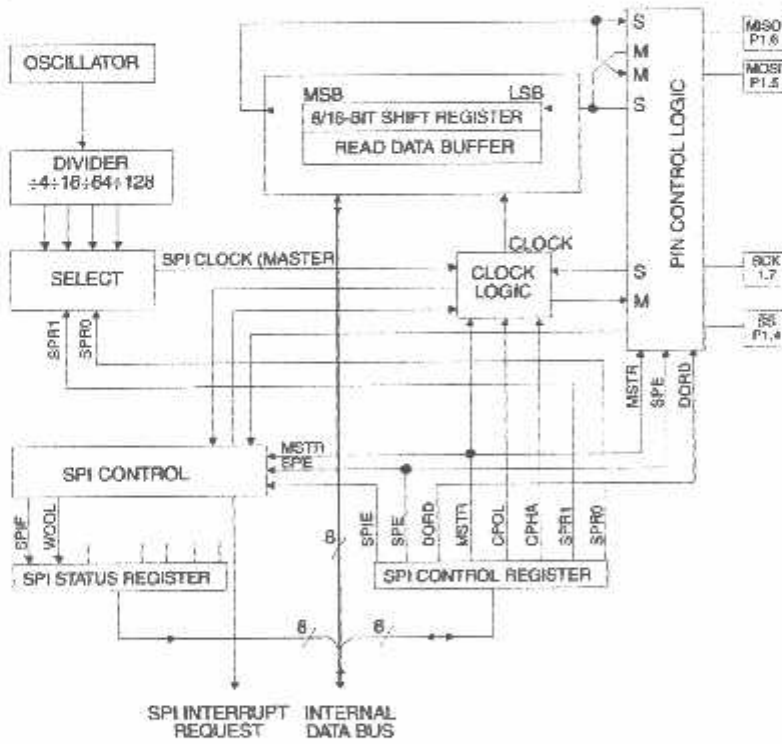
$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (RCAP2H, RCAP2L)]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

### 5. Timer 2 in Clock-out Mode



### 3. SPI Block Diagram



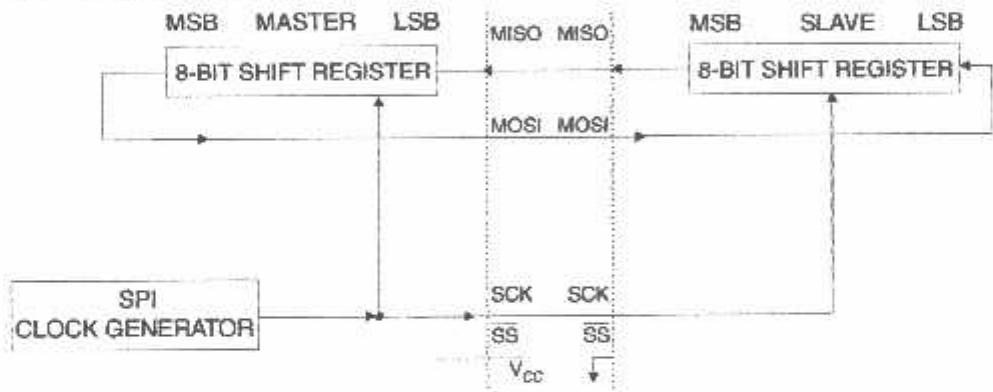
RT in the AT89S8252 operates the same way as RT in the AT89C51, AT89C52 and AT89C55. For information, see the October 1995 Microcontroller book, page 2-49, section titled, "Serial Interface."

I Peripheral Interface

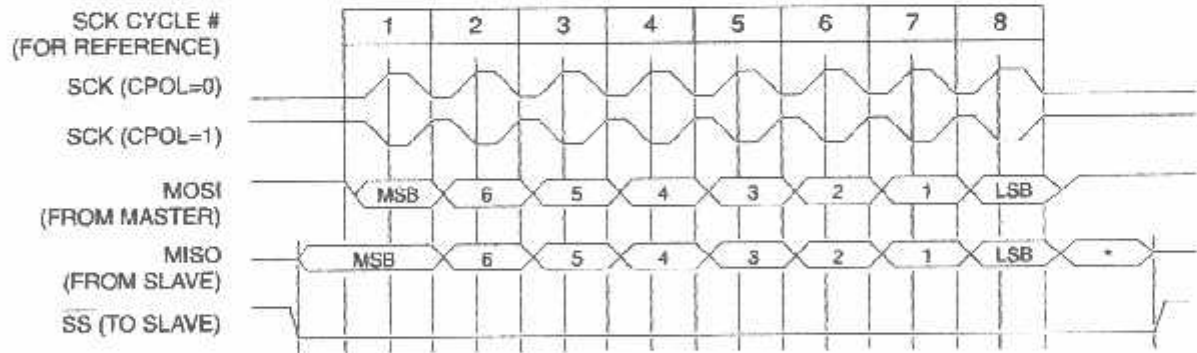
al peripheral interface (SPI) allows high-speed syn- us data transfer between the AT89S8252 and al devices or between several AT89S8252 s. The AT89S8252 SPI features include the

- Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 100 KHz Bit Frequency (max.)
- First or MSB First Data Transfer
- Programmable Bit Rates
- Write Collision Flag Protection

7. SPI Master-slave Interconnection



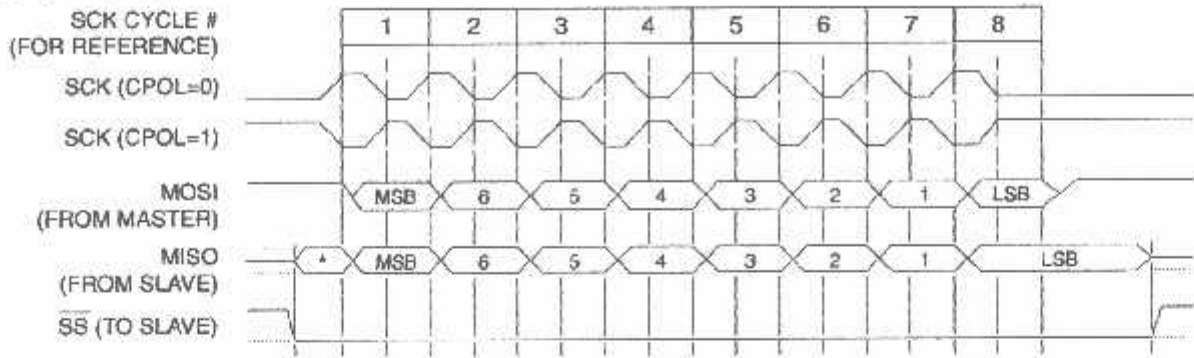
3. SPI transfer Format with CPHA = 0



ined but normally MSB of character just received



### 9. SPI Transfer Format with CPHA = 1



ined but normally LSB of previously transmitted character

### upts

89S8252 has a total of six interrupt vectors: two interrupts (INT0 and INT1), three timer interrupts (0, 1, and 2), and the serial port interrupt. These are all shown in Figure 10.

these interrupt sources can be individually enabled by setting or clearing a bit in Special Function IE. IE also contains a global disable bit, EA, which all interrupts at once.

it Table 10 shows that bit position IE.6 is unimple- l. In the AT89C51, bit position IE.5 is also mented. User software should not write 1s to these tions, since they may be used in future AT89 s.

Interrupt is generated by the logical OR of bits TF2 F2 in register T2CON. Neither of these flags is by hardware when the service routine is vectored ict, the service routine may have to determine it was TF2 or EXF2 that generated the interrupt, bit will have to be cleared in software.

ier 0 and Timer 1 flags, TF0 and TF1, are set at the cycle in which the timers overflow. The values polled by the circuitry in the next cycle. However, er 2 flag, TF2, is set at S2P2 and is polled in the cle in which the timer overflows.

**Table 10. Interrupt Enable (IE) Register**

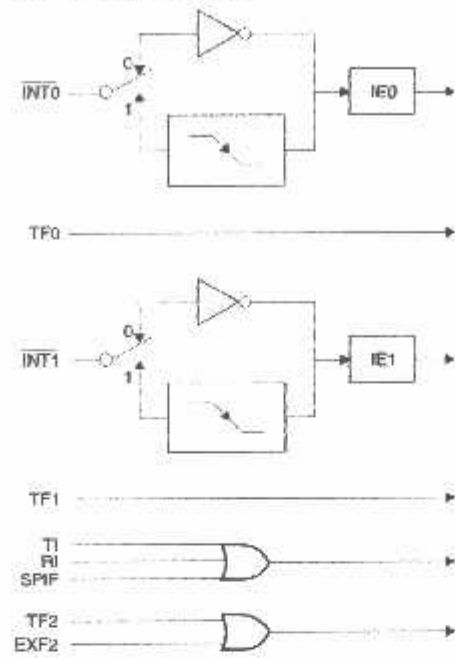
(MSB)(LSB)							
EA	—	ET2	ES	ET1	EX1	ET0	EX0
Enable Bit = 1 enables the interrupt.							
Enable Bit = 0 disables the interrupt.							

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

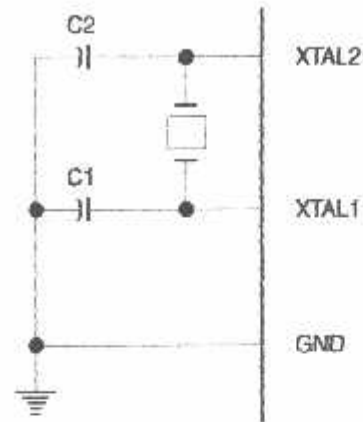
## 10. Interrupt Sources



## Oscillator Characteristics

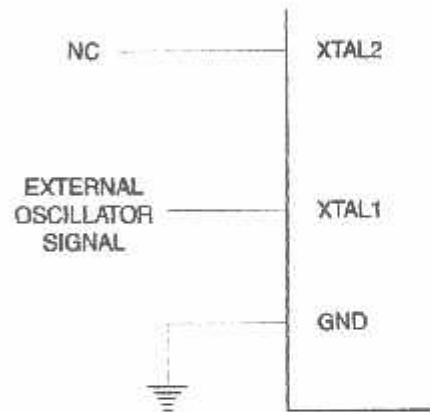
XTAL1 and XTAL2 are the input and output, respectively, of the on-chip inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the oscillator from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be met.

Figure 11. Oscillator Connections



Note: Note: C1, C2 = 30 pF ± 10 pF for Crystals  
= 40 pF ± 10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration





## Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by the `HALT` instruction. The content of the on-chip RAM and all the special function registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

When idle mode is terminated by a hardware reset, the device normally resumes program execution

from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

## States of External Pins During Idle and Power-down Modes

	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
	Internal	1	1	Data	Data	Data	Data
	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

## Power-down Mode

In power-down mode, the oscillator is stopped and the device enters the low-power state. The on-chip RAM and Special Function Registers retain their values until the power-down mode is exited. Exit from power-down can be initiated either by a hardware reset or by an enabled external interrupt. Reset resets the SFRs but does not change the on-chip RAM. The device should not be activated before  $V_{CC}$  is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

When power-down is entered via an interrupt, the external interrupt must be enabled as level sensitive before entering power-down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

## Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the  $\overline{EA}$  pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of  $\overline{EA}$  must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

## Bit Protection Modes<sup>(1)(2)</sup>

Program Lock Bits			Protection Type
LB1	LB2	LB3	
U	U	U	No internal memory lock feature.
P	U	U	MOVX instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
P	P	P	Same as Mode 3, but external execution is also disabled.

1. U = Unprogrammed

2. P = Programmed



## Programming the Flash and EEPROM

AT89S8252 Flash Microcontroller offers 8K bytes of non-volatile reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

AT89S8252 is normally shipped with the on-chip Flash and EEPROM Data memory arrays in the erased state (contents = FFH) and ready to be programmed. It also supports a High-voltage (12V) Parallel programming mode and a Low-voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside a target system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In parallel programming mode, the two arrays occupy one continuous address space: 0000H to 1FFFFH for the Code array and 2000H to 27FFFH for the Data array.

Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode if any of the lock bits have been programmed.

In parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

**Programming Algorithm:** To program and verify AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:
2. Apply power between  $V_{CC}$  and GND pins.
3. Set RST pin to "H".
4. Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait at least 10 milliseconds.
5. Pull  $\overline{PSEN}$  pin to "L".
6. Set  $\overline{EA}$  pin to "H".
7. Set  $\overline{RDY}/\overline{BSY}$  pin to "H" and all other pins to "H".
8. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
9. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
10. Apply data to pins P0.0 to P0.7 for Write Code operation.

5. Raise  $\overline{EA}/V_{PP}$  to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/ $\overline{PROG}$  once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
  - Set XTAL1 to "L".
  - Set RST and  $\overline{EA}$  pins to "L".
  - Turn  $V_{CC}$  power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

**Data Polling:** The AT89S8252 features  $\overline{DATA}$  Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin.  $\overline{DATA}$  Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The progress of byte programming in the parallel programming mode can also be monitored by the  $\overline{RDY}/\overline{BSY}$  output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate  $\overline{BSY}$ . P3.4 is pulled High again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

**Chip Erase:** Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/ $\overline{PROG}$  low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.

erial programming mode, a chip erase operation is by issuing the Chip Erase instruction. In this mode, se is self-timed and takes about 16 ms.

chip erase, a serial read from any address location m 00H at the data outputs.

**Programming Fuse:** A programmable fuse is available to disable Serial Programming if the user needs m system security. The Serial Programming Fuse r be programmed or erased in the Parallel Program-ode.

89S8252 is shipped with the Serial Programming abled.

**g the Signature Bytes:** The signature bytes are the same procedure as a normal verification of s 030H and 031H, except that P3.6 and P3.7 must d to a logic low. The values returned are as follows:  
 1H) = 1EH indicates manufactured by Atmel  
 2H) = 72H indicates 89S8252

## ramming Interface

ode byte in the Flash and EEPROM arrays can be and the entire array can be erased, by using the ate combination of control signals. The write oper-ycle is self-timed and once initiated, will ically time itself to completion.

r programming vendors offer worldwide support for al microcontroller series. Please contact your local ming vendor for the appropriate software revision.

## Downloading

e Code and Data memory arrays can be pro-d using the serial SPI bus while RST is pulled to serial interface consists of pins SCK, MOSI (input) O (output). After RST is set high, the Programming instruction needs to be executed first before pro-ise operations can be executed.

erase cycle is built into the self-timed programming n (in the serial mode ONLY) and there is no need recute the Chip Erase instruction unless any of the have been programmed. The Chip Erase opera-s the content of every memory location in both the d Data arrays into FFH.

le and Data memory arrays have separate address

0000H to 1FFFFH for Code memory and 000H to 7FFFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

## Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:  
 Apply power between VCC and GND pins.  
 Set RST pin to "H".  
 If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):

- Set XTAL1 to "L" (if a crystal is not used).
- Set RST to "L".
- Turn V<sub>CC</sub> power off.

## Serial Programming Instruction

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:






# Instruction Set

Operation	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	x x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

1. DATA polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
2. "aaaaa" = high order address.
3. "x" = don't care.



## and EEPROM Parallel Programming Modes

	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
rog. Modes	H	h <sup>(1)</sup>	h <sup>(1)</sup>	x						
ise	H	L	 <sup>(2)</sup>	12V	H	L	L	L	X	X
0K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
0K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
ck Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
ck Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
nel Code	H	L	H	12V	L	L	L	L	DOUT	30H
vice Code	H	L	H	12V	L	L	L	L	DOUT	31H
og. Enable	H	L	 <sup>(2)</sup>	12V	L	H	L	H	P0.0 = 0	X
og. Disable	H	L	 <sup>(2)</sup>	12V	L	H	L	H	P0.0 = 1	X
rial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

1. "h" = weakly pulled "High" internally.

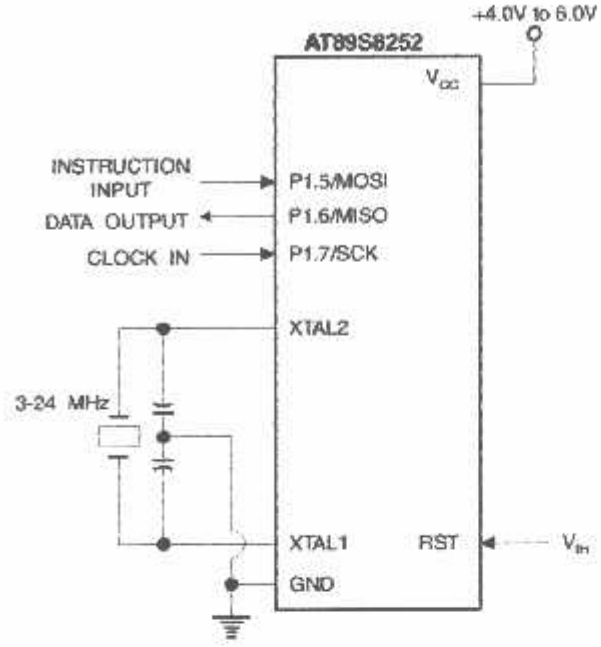
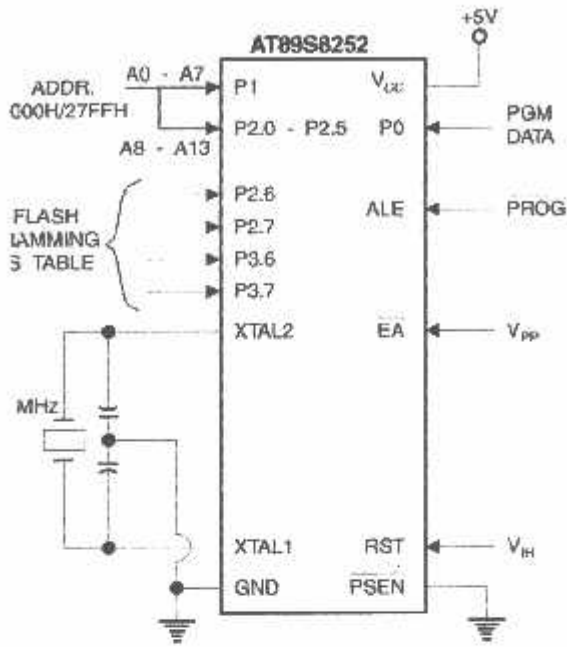
2. Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.

3. P3.4 is pulled Low during programming to indicate RDY/BSY.

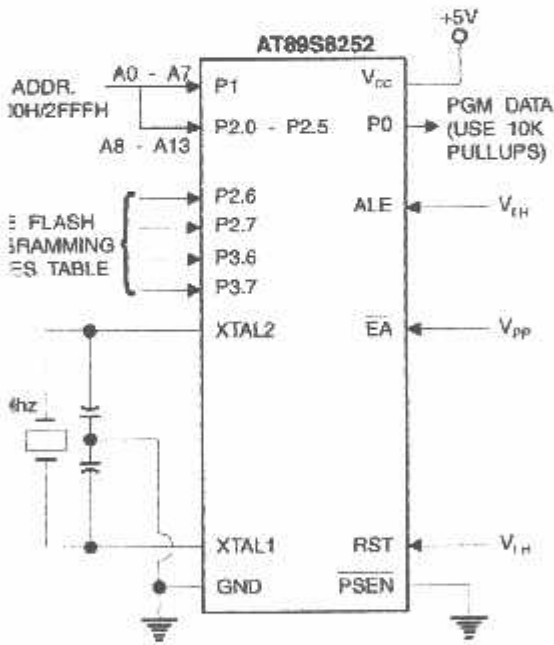
4. "X" = don't care

### 13. Programming the Flash/EEPROM Memory

Figure 15. Flash/EEPROM Serial Downloading



### 4. Verifying the Flash/EEPROM Memory

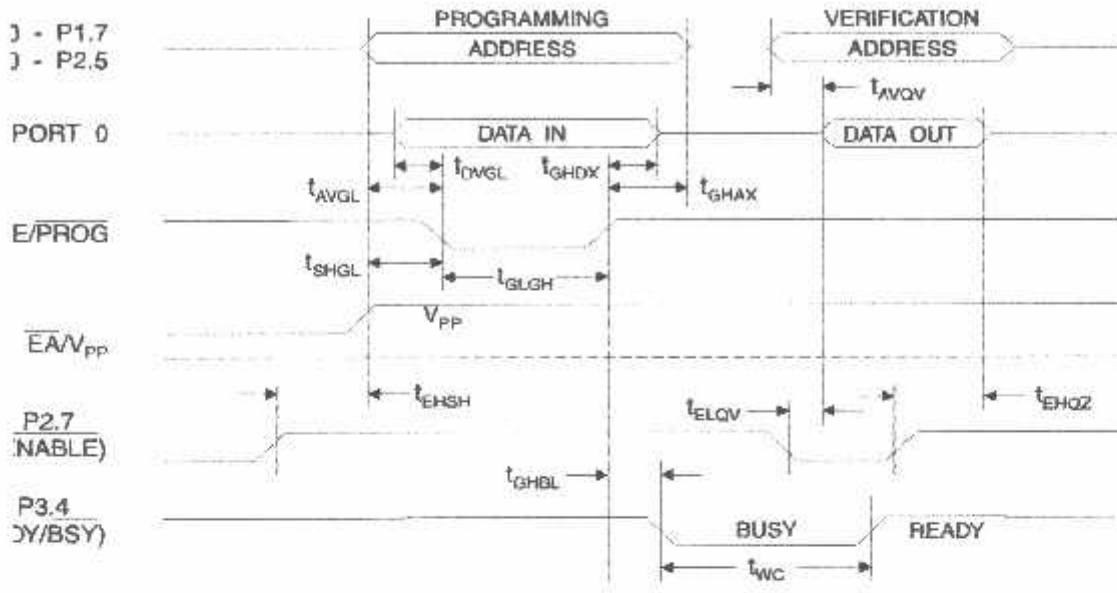


## Programming and Verification Characteristics – Parallel Mode

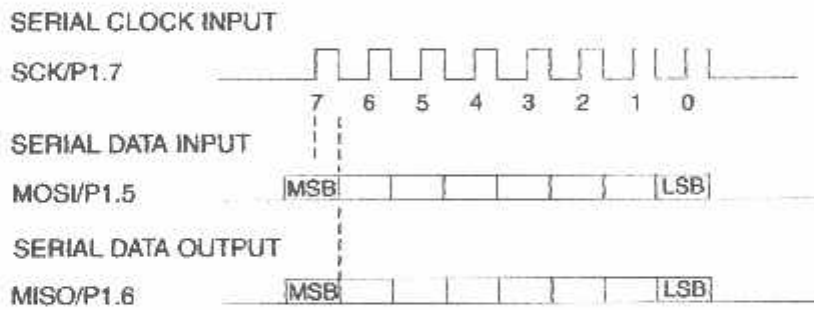
to 70°C,  $V_{CC} = 5.0V \pm 10\%$

Parameter	Min	Max	Units
Programming Enable Voltage	11.5	12.5	V
Programming Enable Current		1.0	mA
Oscillator Frequency	3	24	MHz
Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
Address Hold after $\overline{PROG}$	$48t_{CLCL}$		
Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
Data Hold after $\overline{PROG}$	$48t_{CLCL}$		
P2.7 ( $\overline{ENABLE}$ ) High to $V_{PP}$	$48t_{CLCL}$		
$V_{PP}$ Setup to $\overline{PROG}$ Low	10		$\mu s$
$\overline{PROG}$ Width	1	110	$\mu s$
Address to Data Valid		$48t_{CLCL}$	
$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
Data Float after $\overline{ENABLE}$	0	$48t_{CLCL}$	
$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	$\mu s$
Byte Write Cycle Time		2.0	ms

## /EEPROM Programming and Verification Waveforms – Parallel Mode



## Downloading Waveforms





## Absolute Maximum Ratings\*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
Maximum Output Current	15.0 mA

**\*NOTICE:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Electrical Characteristics

Values shown in this table are valid for  $T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$  and  $V_{CC} = 5.0\text{V} \pm 20\%$ , unless otherwise noted.

Parameter	Condition	Min	Max	Units
Input Low-voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
Input Low-voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
Output Low-voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
Output Low-voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.5	V
Output High-voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
	$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
	$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
	$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
	$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
Reset Pull-down Resistor		50	300	$\text{K}\Omega$
Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^{\circ}\text{C}$		10	pF
Power Supply Current	Active Mode, 12 MHz		25	mA
	Idle Mode, 12 MHz		6.5	mA
Power-down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$		100	$\mu\text{A}$
	$V_{CC} = 3\text{V}$		40	$\mu\text{A}$

- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 10 mA  
Maximum  $I_{OL}$  per 8-bit port:  
Port 0: 26 mA  
Ports 1, 2, 3: 15 mA

Maximum total  $I_{OL}$  for all output pins: 71 mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

- Minimum  $V_{CC}$  for Power-down is 2V



### Characteristics

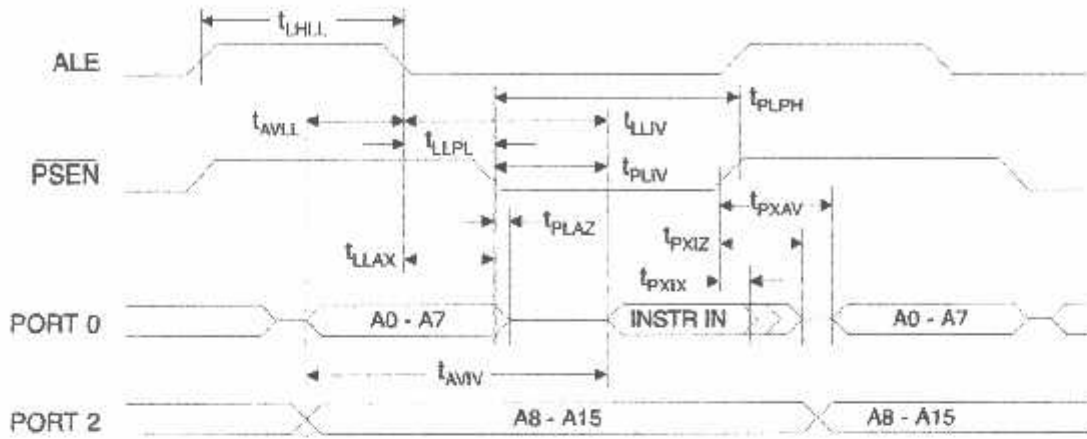
Operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$ , and  $\overline{\text{PSEN}}$  = 100 pF; load capacitance for all other = 80 pF.

### AC Timing Diagrams and Data Memory Characteristics

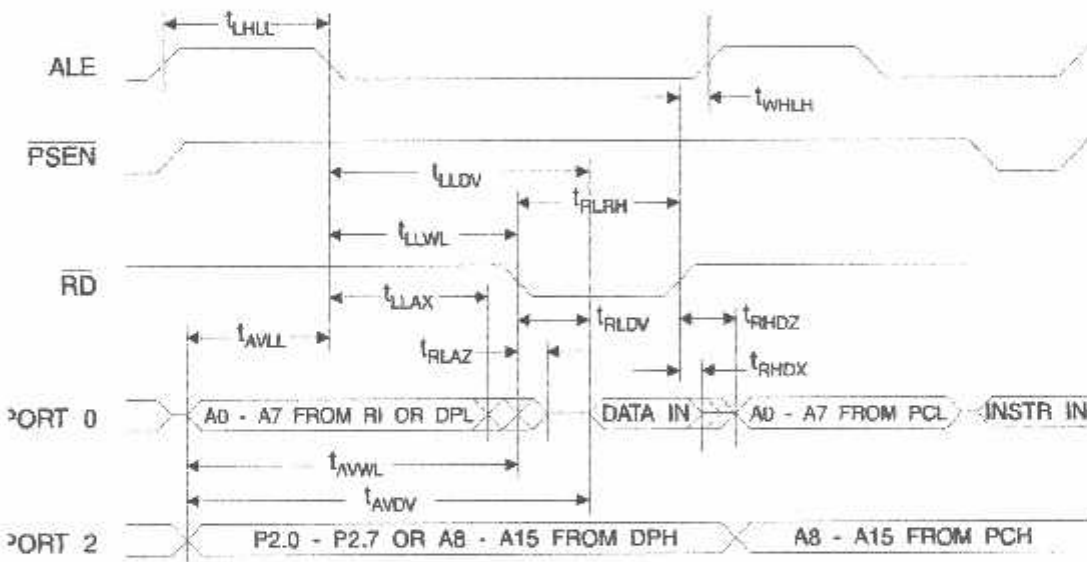
I	Parameter	Variable Oscillator		Units
		Min	Max	
	Oscillator Frequency	0	24	MHz
	ALE Pulse Width	$2t_{\text{CLCL}} - 40$		ns
	Address Valid to ALE Low	$t_{\text{CLCL}} - 13$		ns
	Address Hold after ALE Low	$t_{\text{CLCL}} - 20$		ns
	ALE Low to Valid Instruction In		$4t_{\text{CLCL}} - 65$	ns
	ALE Low to $\overline{\text{PSEN}}$ Low	$t_{\text{CLCL}} - 13$		ns
	$\overline{\text{PSEN}}$ Pulse Width	$3t_{\text{CLCL}} - 20$		ns
	$\overline{\text{PSEN}}$ Low to Valid Instruction In		$3t_{\text{CLCL}} - 45$	ns
	Input Instruction Hold after $\overline{\text{PSEN}}$	0		ns
	Input Instruction Float after $\overline{\text{PSEN}}$		$t_{\text{CLCL}} - 10$	ns
	$\overline{\text{PSEN}}$ to Address Valid	$t_{\text{CLCL}} - 8$		ns
	Address to Valid Instruction In		$5t_{\text{CLCL}} - 55$	ns
	$\overline{\text{PSEN}}$ Low to Address Float		10	ns
	$\overline{\text{RD}}$ Pulse Width	$6t_{\text{CLCL}} - 100$		ns
	$\overline{\text{WR}}$ Pulse Width	$6t_{\text{CLCL}} - 100$		ns
	$\overline{\text{RD}}$ Low to Valid Data In		$5t_{\text{CLCL}} - 90$	ns
	Data Hold after $\overline{\text{RD}}$	0		ns
	Data Float after $\overline{\text{RD}}$		$2t_{\text{CLCL}} - 28$	ns
	ALE Low to Valid Data In		$8t_{\text{CLCL}} - 150$	ns
	Address to Valid Data In		$9t_{\text{CLCL}} - 165$	ns
	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	$3t_{\text{CLCL}} - 50$	$3t_{\text{CLCL}} + 50$	ns
	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	$4t_{\text{CLCL}} - 75$		ns
	Data Valid to $\overline{\text{WR}}$ Transition	$t_{\text{CLCL}} - 20$		ns
	Data Valid to $\overline{\text{WR}}$ High	$7t_{\text{CLCL}} - 120$		ns
	Data Hold after $\overline{\text{WR}}$	$t_{\text{CLCL}} - 20$		ns
	$\overline{\text{RD}}$ Low to Address Float		0	ns
	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	$t_{\text{CLCL}} - 20$	$t_{\text{CLCL}} + 25$	ns



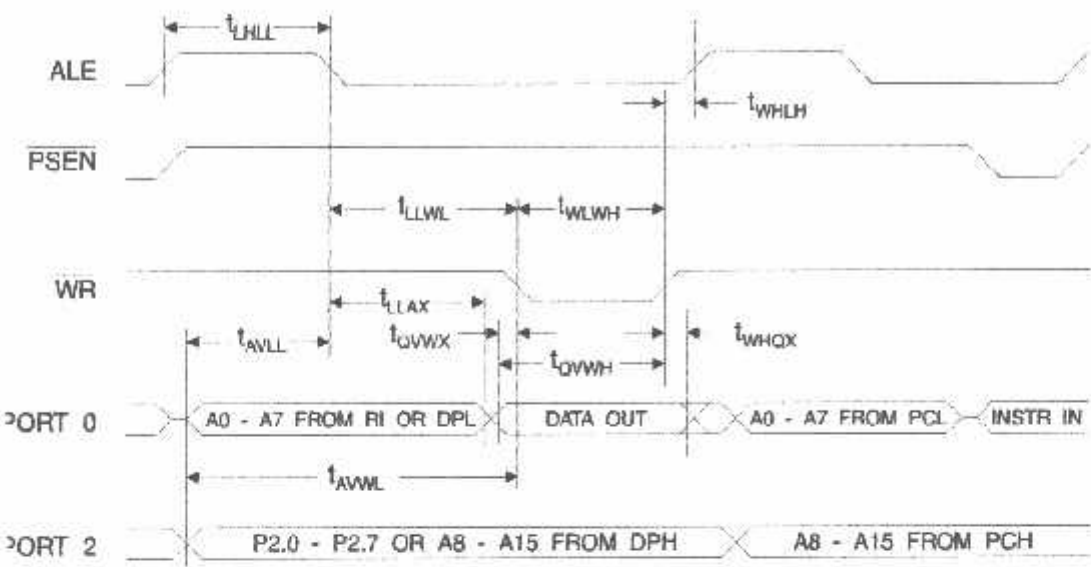
## nal Program Memory Read Cycle



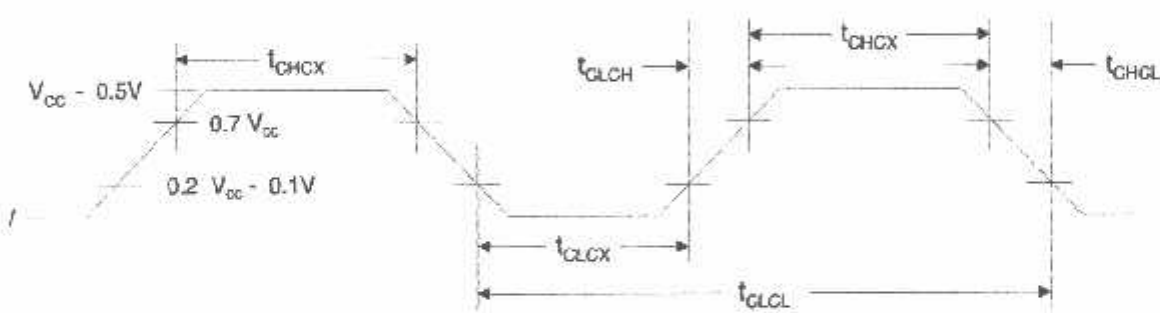
## nal Data Memory Read Cycle



nal Data Memory Write Cycle



nal Clock Drive Waveforms

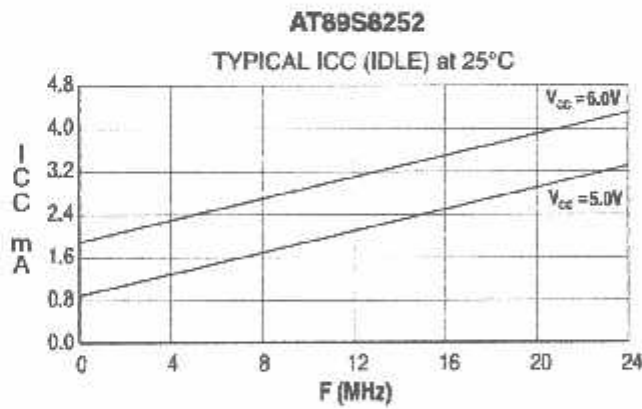
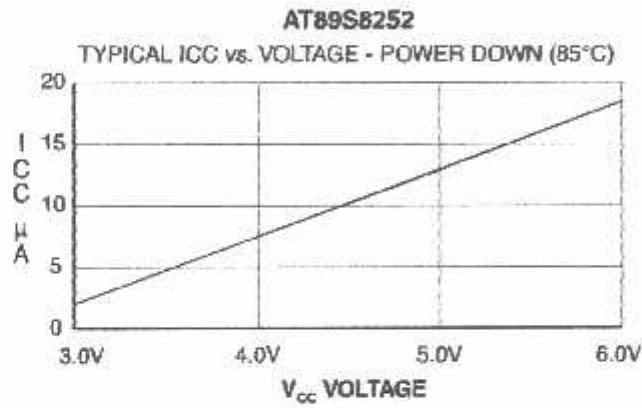
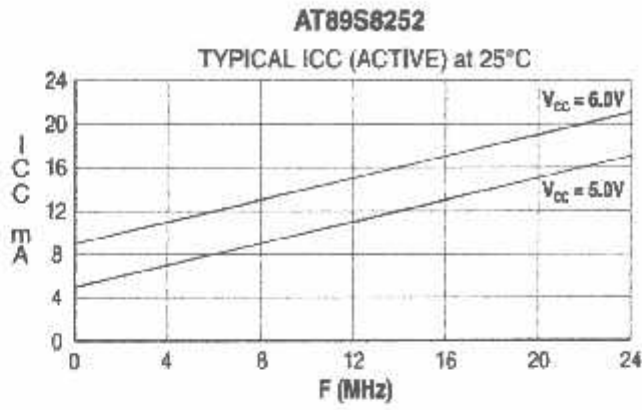


nal Clock Drive

Parameter	V <sub>CC</sub> = 4.0V to 6.0V		Units
	Min	Max	
Oscillator Frequency	0	24	MHz
Clock Period	41.6		ns
High Time	15		ns
Low Time	15		ns
Rise Time		20	ns
Fall Time		20	ns



ies in this table are valid for  $V_{CC} = 4.0V$  to  $6V$  and Load Capacitance =  $80\text{ pF}$ .



- Notes: 1. XTAL1 tied to GND for I<sub>CC</sub> (power-down)  
2. Lock bits programmed



ing Information

d )	Power Supply	Ordering Code	Package	Operation Range
	4.0V to 6.0V	AT89S8252-24AC AT89S8252-24JC AT89S8252-24PC AT89S8252-24QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
	4.0V to 6.0V	AT89S8252-24AI AT89S8252-24JI AT89S8252-24PI AT89S8252-24QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
	4.5V to 5.5V	AT89S8252-33AC AT89S8252-33JC AT89S8252-33PC AT89S8252-33QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)

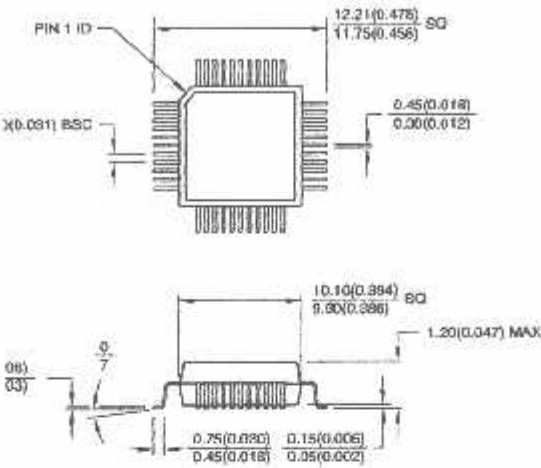
= Preliminary Information

Package Type	
	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
	44-lead, Plastic J-leaded Chip Carrier (PLCC)
	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

AT89S8252

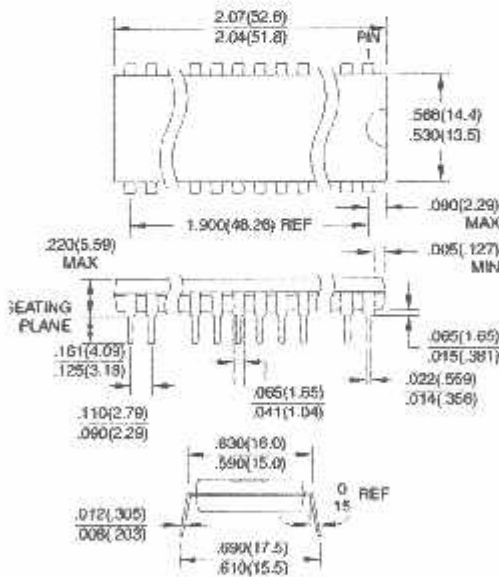
ing Information

, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad  
ack (TQFP)  
ensions in Millimeters and (Inches)\*  
:C STANDARD MS-026 ACB

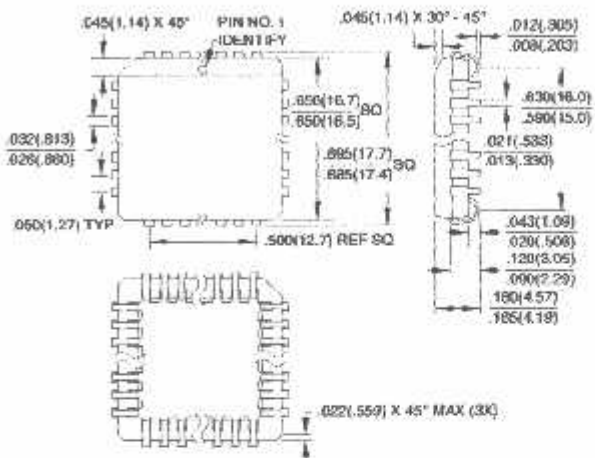


rolling dimension: millimeters

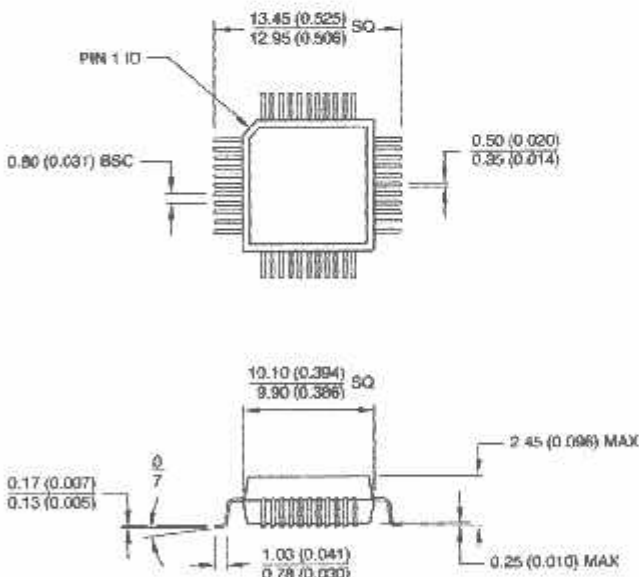
5, 40-lead, 0.600" Wide, Plastic Dual Inline  
age (PDIP)  
ensions in Inches and (Millimeters)



44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)  
Dimensions in Inches and (Millimeters)  
JEDEC STANDARD MS-018 AC



44Q, 44-lead, Plastic Quad Flat Package (PQFP)  
Dimensions in Millimeters and (Inches)\*  
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters

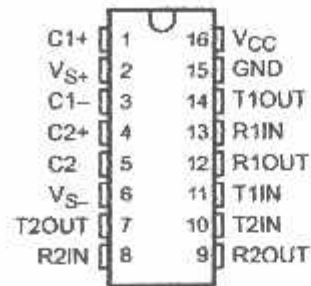


# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- $\pm 30$ -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE  
MAX232I . . . D, DW, OR N PACKAGE  
(TOP VIEW)



## Description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept  $\pm 30$ -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

## ORDERING INFORMATION

T <sub>A</sub>	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
–40°C to 85°C	SOP (NS)	Tape and reel	MAX232NSR	MAX232
	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2002, Texas Instruments Incorporated



232, MAX232I  
- EIA-232 DRIVERS/RECEIVERS

1 - FEBRUARY 1989 - REVISED OCTOBER 2002

Function Tables

EACH DRIVER

INPUT T <sub>IN</sub>	OUTPUT T <sub>OUT</sub>
L	H
H	L

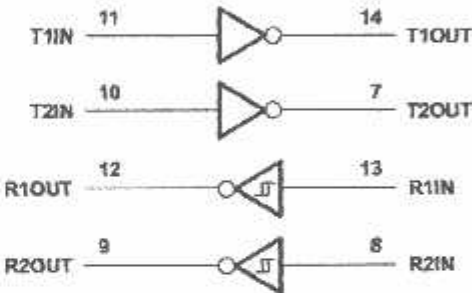
H = high level, L = low level

EACH RECEIVER

INPUT R <sub>IN</sub>	OUTPUT R <sub>OUT</sub>
L	H
H	L

H = high level, L = low level

diagram (positive logic)



POST OFFICE BOX 655363 • DALLAS, TEXAS 75265

# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLL80471—FEBRUARY 1989—REVISED OCTOBER 2002

## Absolute maximum ratings over operating free-air temperature range (unless otherwise noted)<sup>†</sup>

Input supply voltage range, $V_{CC}$ (see Note 1)	−0.3 V to 6 V
Positive output supply voltage range, $V_{S+}$	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, $V_{S-}$	−0.3 V to −15 V
Input voltage range, $V_I$ : Driver	−0.3 V to $V_{CC} + 0.3$ V
Receiver	±30 V
Output voltage range, $V_O$ : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	−0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, $\theta_{JA}$ (see Note 2): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, $T_{stg}$	−65°C to 150°C

Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to network ground terminal.

2. The package thermal impedance is calculated in accordance with JEDEC 51-7.

## Recommended operating conditions

		MIN	NOM	MAX	UNIT
$V_{CC}$	Supply voltage	4.5	5	5.5	V
$V_{IH}$	High-level input voltage (T1IN, T2IN)	2			V
$V_{IL}$	Low-level input voltage (T1IN, T2IN)			0.8	V
T1IN, R2IN	Receiver input voltage			±30	V
A	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	−40	85	

## Electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 3 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP <sup>‡</sup>	MAX	UNIT
$I_{CC}$ Supply current	$V_{CC} = 5.5$ V, All outputs open, $T_A = 25^\circ\text{C}$		8	10	mA

All typical values are at  $V_{CC} = 5$  V and  $T_A = 25^\circ\text{C}$ .

NOTE 3: Test conditions are C1–C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5 \text{ V} \pm 0.5 \text{ V}$ .

DRIVER SECTION

ical characteristics over recommended ranges of supply voltage and operating free-air  
perature range (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
High-level output voltage	T1OUT, T2OUT	$R_L = 3\text{ k}\Omega$ to GND	5	7		V
Low-level output voltage‡	T1OUT, T2OUT	$R_L = 3\text{ k}\Omega$ to GND		–7	–5	V
Output resistance	T1OUT, T2OUT	$V_{S+} = V_{S-} = 0, V_O = \pm 2\text{ V}$	300			$\Omega$
Short-circuit output current	T1OUT, T2OUT	$V_{CC} = 5.5\text{ V}, V_O = 0$		$\pm 10$		mA
Short-circuit input current	T1IN, T2IN	$V_I = 0$			200	$\mu\text{A}$

ical values are at  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$ .  
gebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage  
only.  
re than one output should be shorted at a time.  
: Test conditions are C1–C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

ing characteristics,  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$  (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Driver slew rate		$R_L = 3\text{ k}\Omega$ to 7 k $\Omega$ , See Figure 2			30	V/ $\mu\text{s}$
Driver transition region slew rate		See Figure 3		3		V/ $\mu\text{s}$
Data rate		One TOUT switching		120		kbit/s

: Test conditions are C1–C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

RECEIVER SECTION

ical characteristics over recommended ranges of supply voltage and operating free-air  
perature range (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
High-level output voltage	R1OUT, R2OUT	$I_{OH} = -1\text{ mA}$	3.5			V
Low-level output voltage‡	R1OUT, R2OUT	$I_{OL} = 3.2\text{ mA}$			0.4	V
Receiver positive-going input threshold voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$		1.7	2.4	V
Receiver negative-going input threshold voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$	0.8	1.2		V
Input hysteresis voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}$	0.2	0.5	1	V
Receiver input resistance	R1IN, R2IN	$V_{CC} = 5, T_A = 25^\circ\text{C}$	3	5	7	k $\Omega$

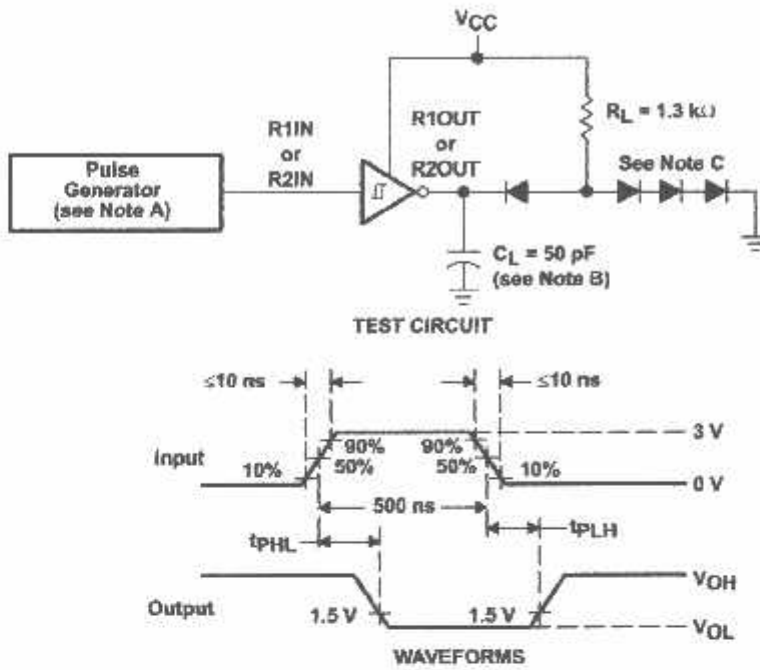
ical values are at  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$ .  
gebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage  
only.  
Test conditions are C1–C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

ing characteristics,  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$  (see Note 3 and Figure 1)

PARAMETER		TYP	UNIT
Receiver propagation delay time, low- to high-level output		500	ns
Receiver propagation delay time, high- to low-level output		500	ns

Test conditions are C1–C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

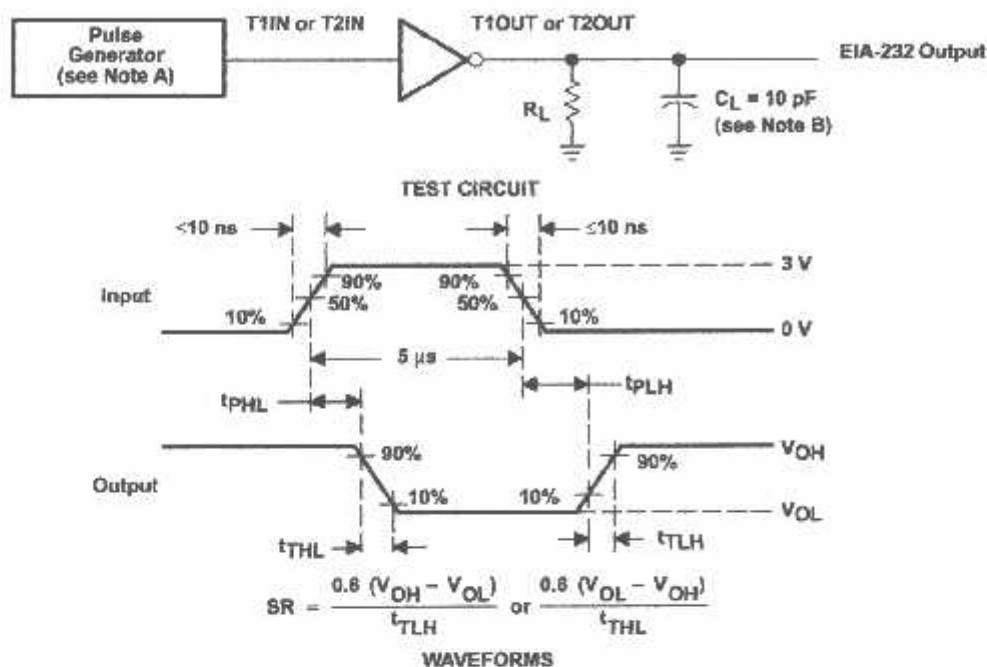
PARAMETER MEASUREMENT INFORMATION



- NOTES: A. The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .  
B.  $C_L$  includes probe and jig capacitance.  
C. All diodes are 1N3064 or equivalent.

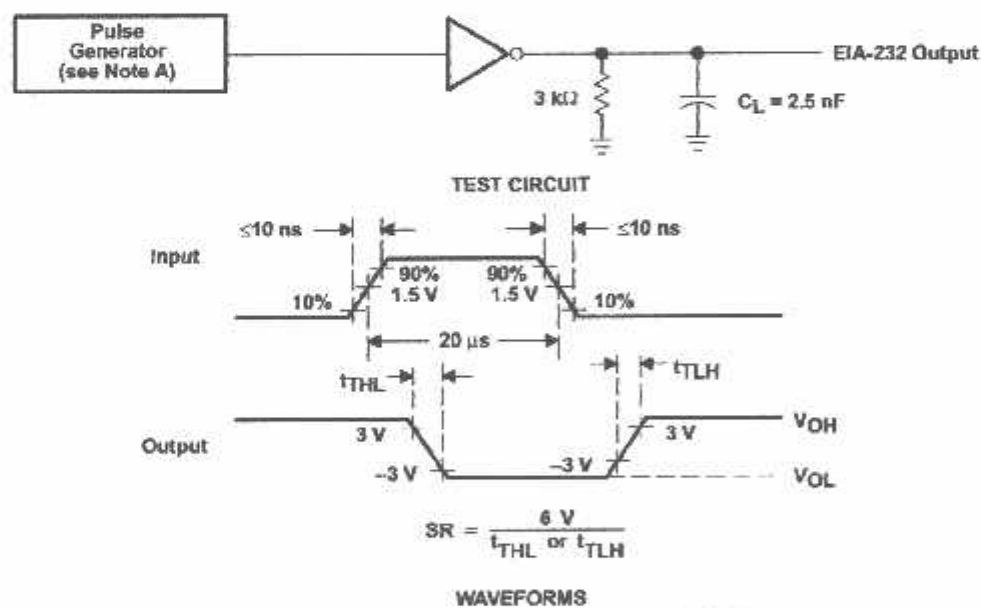
Figure 1. Receiver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements

## PARAMETER MEASUREMENT INFORMATION



- A. The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .  
B.  $C_L$  includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements (5- $\mu\text{s}$  Input)



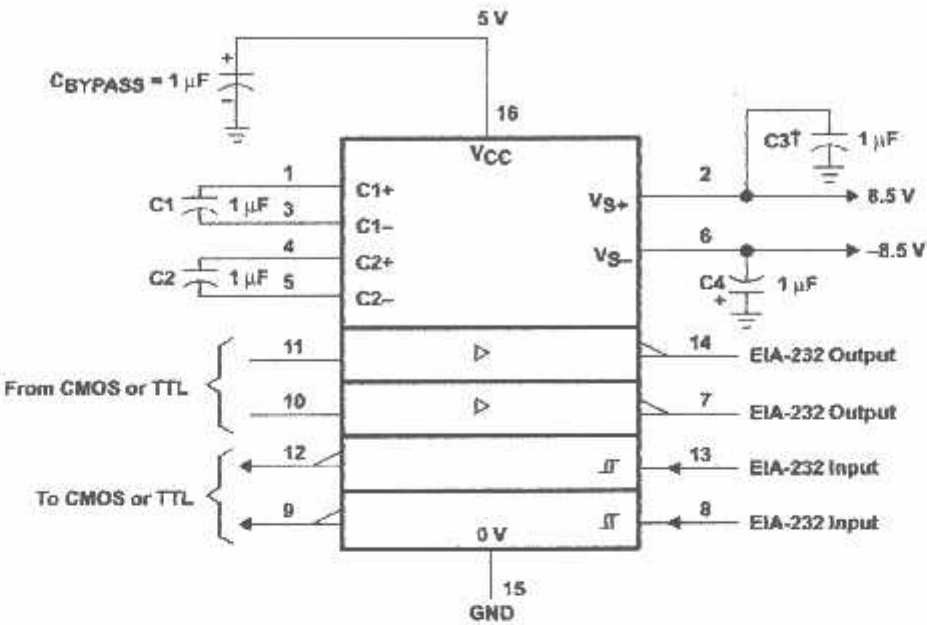
- C. The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .

Figure 3. Test Circuit and Waveforms for  $t_{THL}$  and  $t_{TLH}$  Measurements (20- $\mu\text{s}$  Input)



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

APPLICATION INFORMATION



C3 can be connected to VCC or GND.

Figure 4. Typical Operating Circuit

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

### Mailing Address:

Texas Instruments  
Post Office Box 655303  
Dallas, Texas 75265

22 RI Ring Indicator

Some applications require more pins than a simple async. configurations.

RS-232 Interface

RS-232 (EIA Std.) applicable to the 25 pin interconnection of Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) using serial binary data

Pin	Description	EIA CKT	From DCE	To DCE
1	Frame Ground	AA		
2	Transmitted Data	BA		D (Data)
3	Received Data	BB	D	
4	Request to Send	CA		C (Control)
5	Clear to Send	CB	C	
6	Data Ser Ready	CC	C	
7	Signal Gnd Common Return	AB		
8	Rcvd Line Signal Detector	CF	C	
11	Undefined			
12	Secondary Rcvd Line Sig Detector	SCF	C	
13	Secondary Clear to Send	SCB	C	
14	Secondary Transmitted Data	SBA		D
15	Transmitter Sig. Element Timing	DB	T (Timing)	
16	Secondary Received Data	SBB	D	
17	Receiver Sig. Element Timing	DD	T	
18	Undefined			
19	Secondary Request to Send	SCA		C
20	Data Terminal Ready	CD		C
21	Sig. Quality Detector	CG		C
22	Ring Indicator	CE	C	
23	Data Sig. Rate Selector (DCE)	CI	C	
23	Data Sig. Rate Selector (DTE)	CH		C
24	Transmitter Sig. Element Timing	DA		T
25	Undefined			

Pins used for Synchronous data

[jump to Other Connector pages](#)

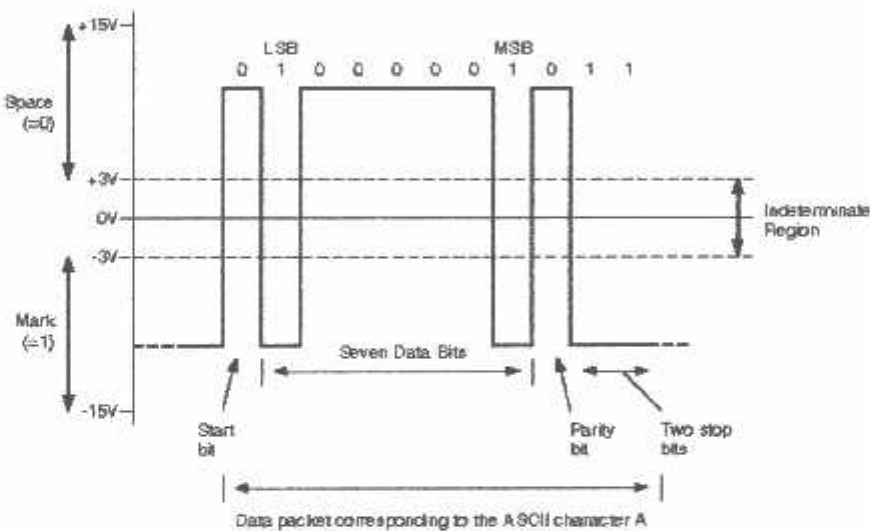
RS-232 Specs.

SPECIFICATIONS		RS232	RS423
Mode of Operation		SINGLE -ENDED	SINGLE -ENDED
Total Number of Drivers and Receivers on One Line		1 DRIVER 1 RECVR	1 DRIVER 10 RECVR
Maximum Cable Length		50 FT.	4000 FT.
Maximum Data Rate		20kb/s	100kb/s



Maximum Driver Output Voltage		+/-25V	+/-6V
Driver Output Signal Level (Loaded Min.)	Loaded	+/-5V to +/-15V	+/-3.6V
Driver Output Signal Level (Unloaded Max)	Unloaded	+/-25V	+/-6V
Driver Load Impedance (Ohms)		3k to 7k	>=450
Max. Driver Current in High Z State	Power On	N/A	N/A
Max. Driver Current in High Z State	Power Off	+/-6mA @ +/-2v	+/-100uA
Slew Rate (Max.)		30V/uS	Adjustable
Receiver Input Voltage Range		+/-15V	+/-12V
Receiver Input Sensitivity		+/-3V	+/-200mV
Receiver Input Resistance (Ohms)		3k to 7k	4k min.

One byte of async data



considerations - you should use cabling made for RS-232 data but I have seen low speed data go over 250' on 2 wire cable. Level 5 cable can also be used but for best distance use a low capacitance data grade cable.

standard maxim length is 50' but if data is async you can increase that distance to as much as 500' with a good grade

RS-232 signal on a single cable is impossible to screen effectively for noise. By screening the entire cable we can reduce the influence of outside noise, but internally generated noise remains a problem. As the baud rate and line length increase, the effect of capacitance between the different lines introduces serious crosstalk (this especially true on synchronous data - because of the clock lines) until a point is reached where the data itself is unreadable. Signal Crosstalk is reduced by using low capacitance cable and shielding each pair

high grade cable (individually shield low capacitance pairs) the distance can be extended to 4000'

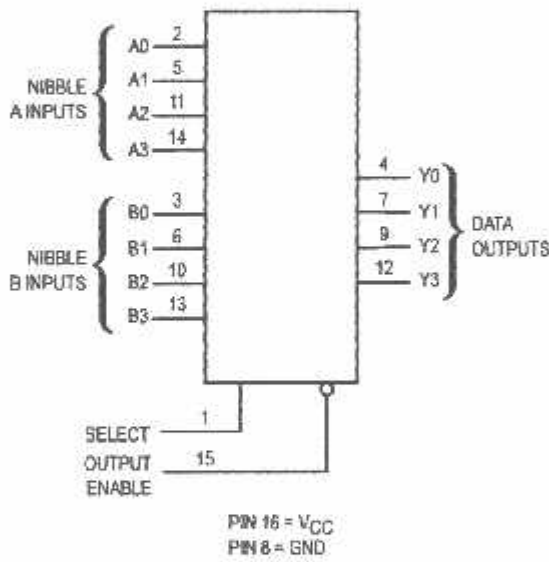
at higher frequencies a new problem comes to light. The high frequency component of the data signal is lost as the cable length increases resulting in a rounded, rather than square wave signal.

Maximum distance will depend on the speed and noise level around the cable run.

# Quad 2-Input Data Selectors/Multiplexers High-Performance Silicon-Gate CMOS

- MC54/74HC157A is identical in pinout to the LS157. The device is compatible with standard CMOS outputs; with pullup resistors, it is compatible with LSTTL outputs.
- The device routes 2 nibbles (A or B) to a single port (Y) as determined by select input. The data is presented at the outputs in noninverted form. A low level on the Output Enable input sets all four Y outputs to a low level.
- MC54/74HC157A is similar in function to the HC257 which has 3-state outputs.
- Output Drive Capability: 10 LSTTL Loads
- Inputs Directly Interface to CMOS, NMOS, and TTL
- Operating Voltage Range: 2.0 to 6.0 V
- Input Current: 1.0  $\mu$ A
- High Noise Immunity Characteristic of CMOS Devices
- Compliance with the Requirements Defined by JEDEC Standard 7A
- Pin Complexity: 82 FETs or 20.5 Equivalent Gates

LOGIC DIAGRAM



## MC54/74HC157A



J SUFFIX  
CERAMIC PACKAGE  
CASE 620-10



N SUFFIX  
PLASTIC PACKAGE  
CASE 648-08



D SUFFIX  
SOIC PACKAGE  
CASE 751B-05



DT SUFFIX  
TSSOP PACKAGE  
CASE 948F-01

ORDERING INFORMATION

MC54HCXXXAJ	Ceramic
MC74HCXXXAN	Plastic
MC74HCXXXAD	SOIC
MC74HCXXXADT	TSSOP

PIN ASSIGNMENT

SELECT	1	16	VCC
A0	2	15	OUTPUT ENABLE
B0	3	14	A3
Y0	4	13	B3
A1	5	12	Y3
B1	6	11	A2
Y1	7	10	B2
GND	8	9	Y2

FUNCTION TABLE

Inputs		Outputs Y0 - Y3
Output Enable	Select	
H	X	L
L	L	A0 - A3
L	H	B0 - B3

X = don't care  
A0 - A3, B0 - B3 = the levels of the respective Data-Word Inputs.

## 54/74HC157A

## MUM RATINGS\*

Parameter	Value	Unit
DC Supply Voltage (Referenced to GND)	- 0.5 to + 7.0	V
DC Input Voltage (Referenced to GND)	- 0.5 to $V_{CC} + 0.5$	V
DC Output Voltage (Referenced to GND)	- 0.5 to $V_{CC} + 0.5$	V
DC Input Current, per Pin	$\pm 20$	mA
DC Output Current, per Pin	$\pm 25$	mA
DC Supply Current, $V_{CC}$ and GND Pins	$\pm 50$	mA
Power Dissipation in Still Air, Plastic or Ceramic DIP†	750	mW
SOIC Package†	500	
TSSOP Package†	450	
Storage Temperature	- 65 to + 150	°C
Lead Temperature, 1 mm from Case for 10 Seconds (Plastic DIP, SOIC or TSSOP Package)	260	°C
(Ceramic DIP)	300	

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation,  $V_{in}$  and  $V_{out}$  should be constrained to the range  $GND \leq (V_{in} \text{ or } V_{out}) \leq V_{CC}$ . Unused inputs must always be tied to an appropriate logic voltage level (e.g., either GND or  $V_{CC}$ ). Unused outputs must be left open.

\*MUM Ratings are those values beyond which damage to the device may occur. Normal operation should be restricted to the Recommended Operating Conditions.

† — Plastic DIP: - 10 mW/°C from 85° to 125°C  
 Ceramic DIP: - 10 mW/°C from 100° to 125°C  
 SOIC Package: - 7 mW/°C from 85° to 125°C  
 TSSOP Package: - 8.1 mW/°C from 85° to 125°C

†h frequency or heavy load considerations, see Chapter 2 of the Motorola High-Speed CMOS Data Book (DL129/D).

## RECOMMENDED OPERATING CONDITIONS

sol	Parameter	Min	Max	Unit	
c  out	DC Supply Voltage (Referenced to GND)	2.0	6.0	V	
	DC Input Voltage, Output Voltage (Referenced to GND)	0	V <sub>CC</sub>	V	
	Operating Temperature, All Package Types	- 55	+ 125	°C	
f	Input Rise and Fall Time (Figure 1)	V <sub>CC</sub> = 2.0 V V <sub>CC</sub> = 4.5 V V <sub>CC</sub> = 6.0 V	0 0 0	1000 500 400	ns

## ELECTRICAL CHARACTERISTICS (Voltages Referenced to GND)

Parameter	Test Conditions	$V_{CC}$ V	Guaranteed Limit			Unit
			- 55 to 25°C	< 85°C	< 125°C	
Minimum High-Level Input Voltage	$V_{out} = V_{CC} - 0.1 \text{ V}$ $ I_{out}  \leq 20 \mu\text{A}$	2.0 3.0 4.5 6.0	1.5 2.1 3.15 4.2	1.5 2.1 3.15 4.2	1.5 2.1 3.15 4.2	V
Maximum Low-Level Input Voltage	$V_{out} = 0.1 \text{ V}$ $ I_{out}  \leq 20 \mu\text{A}$	2.0 3.0 4.5 6.0	0.5 0.9 1.35 1.8	0.5 0.9 1.35 1.8	0.5 0.9 1.35 1.8	V
Minimum High-Level Output Voltage	$V_{in} = V_{IH}$ $ I_{out}  \leq 20 \mu\text{A}$	2.0 4.5 6.0	1.9 4.4 5.9	1.9 4.4 5.9	1.9 4.4 5.9	V
	$V_{in} = V_{IH}$ $ I_{out}  \leq 2.4 \text{ mA}$ $ I_{out}  \leq 6.0 \text{ mA}$ $ I_{out}  \leq 7.8 \text{ mA}$	3.0 4.5 6.0	2.48 3.98 5.48	2.34 3.84 5.34	2.2 3.7 5.2	V

**ELECTRICAL CHARACTERISTICS** (Voltages Referenced to GND)

Symbol	Parameter	Test Conditions	V <sub>CC</sub> V	Guaranteed Limit			Unit
				-55 to 25°C	≤ 85°C	≤ 125°C	
V <sub>OL</sub>	Maximum Low-Level Output Voltage	V <sub>in</sub> = V <sub>IL</sub>  I <sub>out</sub>   ≤ 20 μA	2.0	0.1	0.1	0.1	V
			4.5	0.1	0.1	0.1	
			6.0	0.1	0.1	0.1	
		V <sub>in</sub> = V <sub>IL</sub>  I <sub>out</sub>   ≤ 2.4 mA  I <sub>out</sub>   ≤ 6.0 mA  I <sub>out</sub>   ≤ 7.8 mA	3.0	0.26	0.33	0.4	
			4.5	0.26	0.33	0.4	
			6.0	0.26	0.33	0.4	
I <sub>Y</sub>	Maximum Input Leakage Current	V <sub>in</sub> = V <sub>CC</sub> or GND	6.0	± 0.1	± 1.0	± 1.0	μA
I <sub>Z</sub>	Maximum Three-State Leakage Current	Output in High-Impedance State V <sub>in</sub> = V <sub>IL</sub> or V <sub>IH</sub> V <sub>out</sub> = V <sub>CC</sub> or GND	6.0	± 0.5	± 5.0	± 10	μA
I <sub>C</sub>	Maximum Quiescent Supply Current (per Package)	V <sub>in</sub> = V <sub>CC</sub> or GND I <sub>out</sub> = 0 μA	6.0	4.0	40	160	μA

Information on typical parametric values can be found in Chapter 2 of the Motorola High-Speed CMOS Data Book (DL129/D).

**ELECTRICAL CHARACTERISTICS** (C<sub>L</sub> = 50 pF, Input t<sub>r</sub> = t<sub>f</sub> = 6.0 ns)

Symbol	Parameter	V <sub>CC</sub> V	Guaranteed Limit			Unit
			-55 to 25°C	≤ 85°C	≤ 125°C	
t <sub>PL</sub>	Maximum Propagation Delay, Input A or B to Output Y (Figures 1 and 4)	2.0	105	130	160	ns
		3.0	65	85	115	
		4.5	21	26	32	
		6.0	18	22	27	
t <sub>HL</sub>	Maximum Propagation Delay, Select to Output Y (Figures 2 and 4)	2.0	110	140	185	ns
		3.0	70	90	115	
		4.5	22	28	33	
		6.0	19	24	28	
t <sub>EL</sub>	Maximum Propagation Delay, Output Enable to Output Y (Figures 3 and 4)	2.0	100	125	150	ns
		3.0	80	80	110	
		4.5	20	25	30	
		6.0	17	21	26	
t <sub>HL</sub>	Maximum Output Transition Time, Any Output (Figures 1 and 4)	2.0	75	95	110	ns
		3.0	27	32	36	
		4.5	15	19	22	
		6.0	13	16	19	
C <sub>in</sub>	Maximum Input Capacitance	—	10	10	10	pF

For propagation delays with loads other than 50 pF, and information on typical parametric values, see Chapter 2 of the Motorola High-Speed CMOS Data Book (DL129/D).

P <sub>D</sub>	Power Dissipation Capacitance (Per Package)*	Typical @ 25°C, V <sub>CC</sub> = 5.0 V	pF
		33	

\*to determine the no-load dynamic power consumption:  $P_D = C_{PD} V_{CC}^2 f + I_{CC} V_{CC}$ . For load considerations, see Chapter 2 of the Motorola High-Speed CMOS Data Book (DL129/D).

PIN DESCRIPTIONS

S

I, A2, A3 (Pins 2, 5, 11, 14)

ie A inputs. The data present on these pins is trans- to the outputs when the Select input is at a low level ie Output Enable input is at a low level. The data is ited to the outputs in noninverted form.

I, B2, B3 (Pins 3, 6, 10, 13)

ie B inputs. The data present on these pins is trans- to the outputs when the Select input is at a high level ie Output Enable input is at a low level. The data is ited to the outputs in noninverted form.

UTS

, Y2, Y3 (Pins 4, 7, 9, 12)

i outputs. The selected input Nibble is presented at

these outputs when the Output Enable input is at a low level. The data present on these pins is in its noninverted form. For the Output Enable input at a high level, the outputs are at a low level.

CONTROL INPUTS

Select (Pin 1)

Nibble select. This input determines the data word to be transferred to the outputs. A low level on this input selects the A inputs and a high level selects the B inputs.

Output Enable (Pin 15)

Output Enable input. A low level on this input allows the selected input data to be presented at the outputs. A high level on this input sets all outputs to a low level.

SWITCHING WAVEFORMS

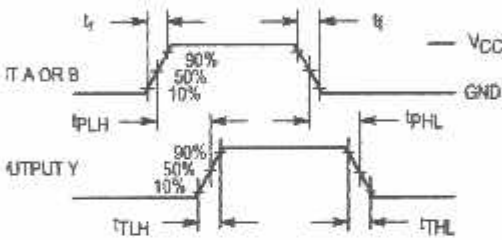


Figure 1. HC157A

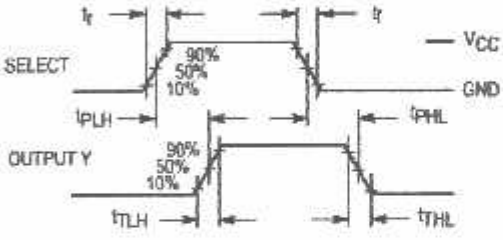


Figure 2. Y versus Select, Noninverted

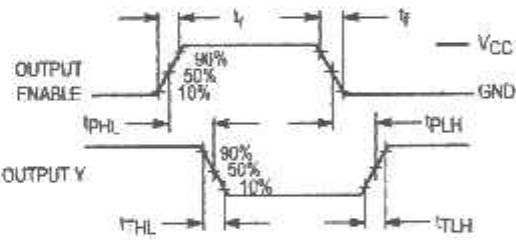
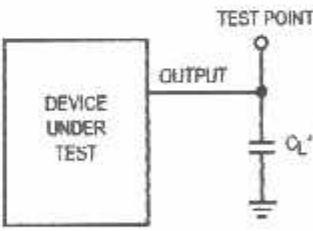
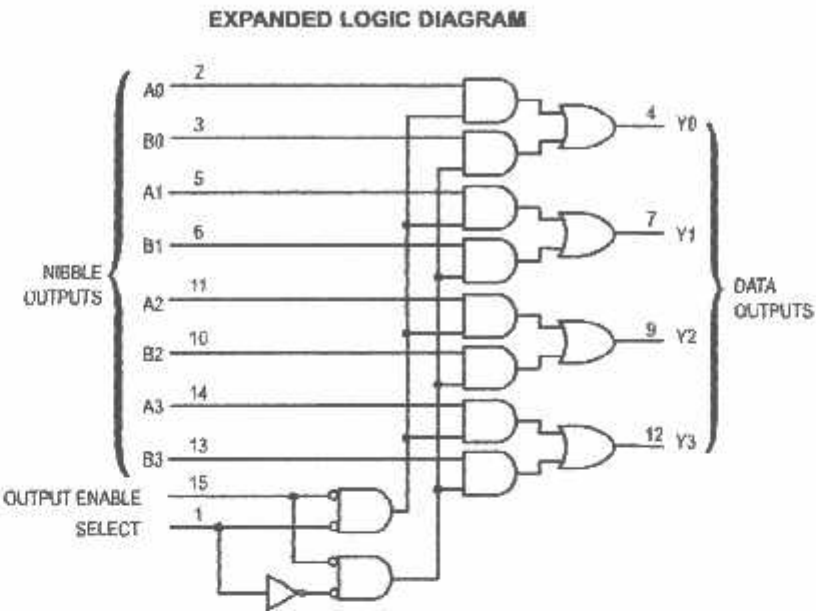


Figure 3. HC157A



\* Includes all probe and jig capacitance

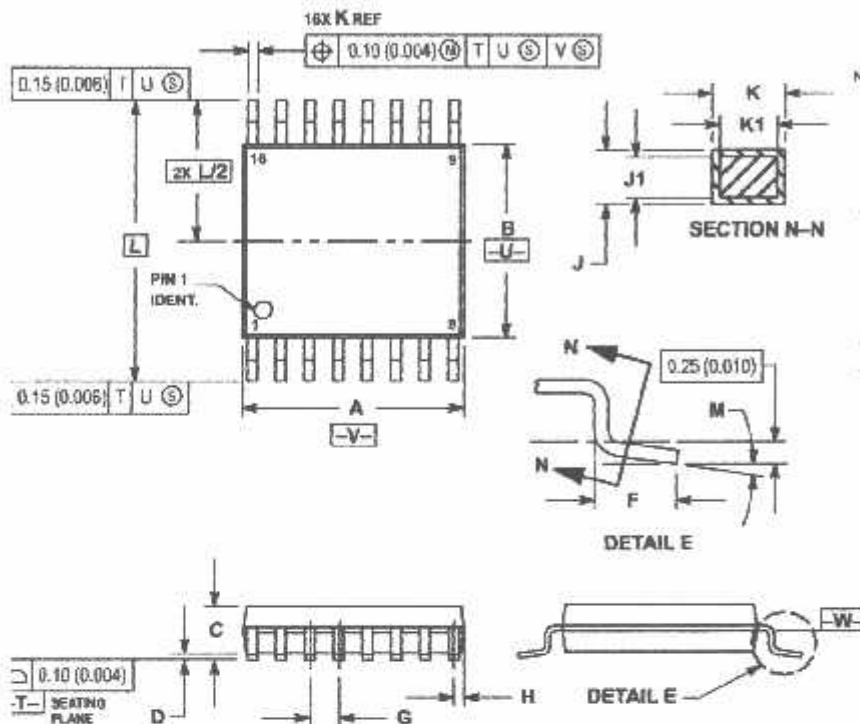
Figure 4. Test Circuit





## OUTLINE DIMENSIONS

DT SUFFIX  
PLASTIC TSSOP PACKAGE  
CASE 948F-01  
ISSUE O



## NOTES

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS. MOLD FLASH OR GATE BURRS SHALL NOT EXCEED 0.15 (0.006) PER SIDE.
4. DIMENSION B DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION. INTERLEAD FLASH OR PROTRUSION SHALL NOT EXCEED 0.25 (0.010) PER SIDE.
5. DIMENSION K DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE K DIMENSION AT MAXIMUM MATERIAL CONDITION.
6. TERMINAL NUMBERS ARE SHOWN FOR REFERENCE ONLY.
7. DIMENSION A AND B ARE TO BE DETERMINED AT DATUM PLANE -W-.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	4.93	5.10	0.193	0.200
B	4.37	4.50	0.172	0.177
C	—	1.20	—	0.047
D	0.05	0.15	0.002	0.006
E	0.50	0.75	0.020	0.030
F	0.65 BSC	0.026 BSC		
G	0.10	0.26	0.004	0.011
H	0.02	0.20	0.001	0.008
J	0.03	0.16	0.001	0.006
K	0.19	0.30	0.007	0.012
K1	0.10	0.25	0.004	0.010
L	0.40 BSC	0.252 BSC		
M	0.10	0.10	0.004	0.004

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding liability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and expressly disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" are validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola, its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees. In no event shall Motorola be liable for any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and M are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

Mifex is a trademark of Motorola, Inc.

## To reach us:

EUROPE/Locations Not Listed: Motorola Literature Distribution;  
Box 5405, Denver, Colorado 80217. 303-675-2140 or 1-800-441-2447

JAPAN: Nippon Motorola Ltd., Tatsumi-SPD-JLDC, 6F Seibu-Butsuryu-Center,  
3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 81-3-3521-8315

RMFAX0@email.sps.mot.com - TOUCHTONE 602-214-6600  
- US & Canada ONLY 1-800-774-1848

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,  
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26022590

INET: <http://www.mot.com/SPS/>

MOTOROLA

MC74HC157AD



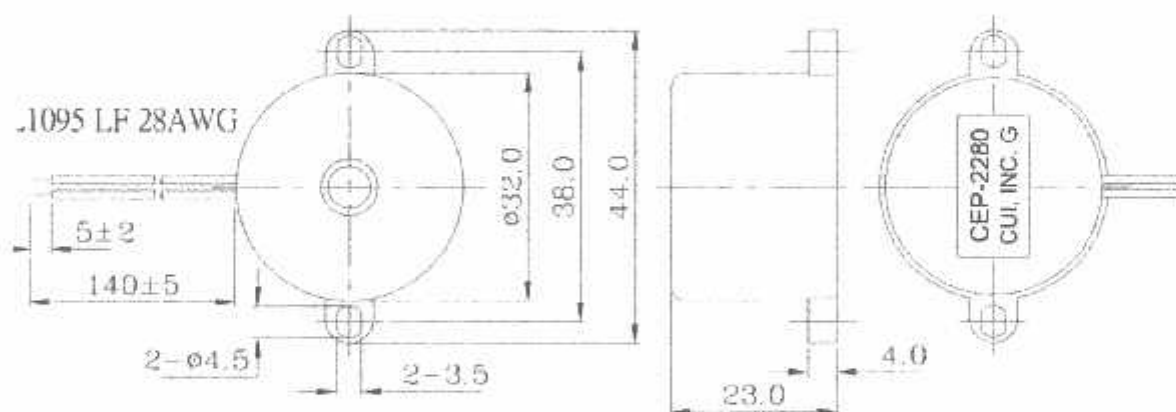


### Specifications

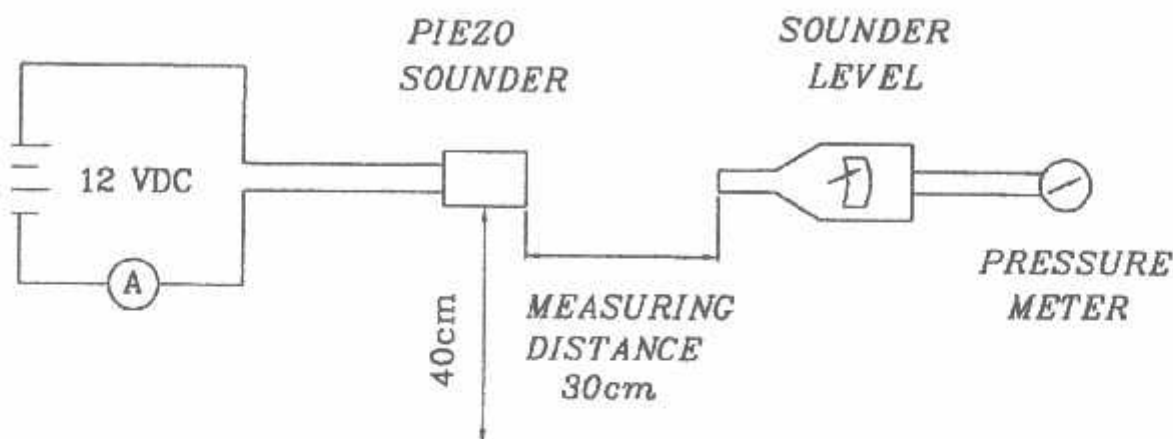
Operating frequency	2.5 ~ 3.8 KHz	
Operating voltage	6.0 ~ 18.0 V dc	
Current consumption	17 mA max.	at 12 V dc
Pressure level	88 db min.	at 30 cm / 12 V dc
Operating voltage	12 V dc	
	High/Low	
Operating temperature	-20 ~ +60° C	
Storage temperature	-30 ~ +70° C	
Dimensions	ø32.0 x H23.0 mm	See attached drawing
	13.20 g max.	
	ABS UL-94 1/16" HB (Black)	
	Wire type	See attached drawing
	yes	

### Reference Drawing

Tolerance: ±0.5



## Measurement Method

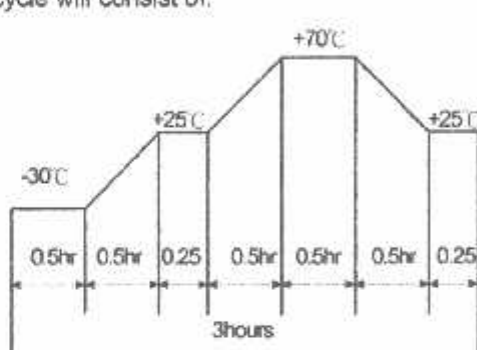


Measuring Circuit  
 RION S.P.L. meter UC 30 or equivalent  
 Hewlett Packard 33120A Function Generator or equivalent

## Mechanical Characteristics

	Test Condition	Evaluation Standard
Reliability (or excepted)	Stripped wires of lead wires are immersed in rosin for 5 seconds and then immersed in a solder bath of $+270 \pm 5^\circ\text{C}$ for $3 \pm 0.5$ seconds.	90% min. of the stripped wires will be wet with solder. (Except the edge of the terminal)
Mechanical Strength	The pull force should be applied to the double lead wire: Horizontal 3.0N (0.306kg) for 30 seconds Vertical 2.0N (0.204kg) for 30 seconds	No damage or cutting off.
Vibration	The buzzer will be measured after applying a vibration amplitude of 1.5 mm with 10 to 55 Hz band of vibration frequency to each of the 3 perpendicular directions for 2 hours.	The value of oscillation frequency/current consumption should be $\pm 10\%$ of the initial measurements. The SPL should be within $\pm 10\text{dB}$ compared with the initial measurement.
Drop Test	The part will be dropped from a height of 75 cm onto a 40 mm thick wooden board 3 times in 3 axis (X, Y, Z) for a total of 9 drops.	

## Environment Test

	Test Condition	Evaluation Standard
p. test	After being placed in a chamber at +70°C for 240 hours.	The buzzer will be measured after being placed at +25°C for 4 hours. The value of the oscillation frequency/current consumption should be within ±10% compared to the initial measurements. The SPL should be within ±10dB compared to the initial measurements.
c. test	After being placed in a chamber at -30°C for 240 hours.	
test	After being placed in a chamber at +40°C and 90±5% relative humidity for 240 hours.	
cle test	The part shall be subjected to 5 cycles. One cycle will consist of:	
		

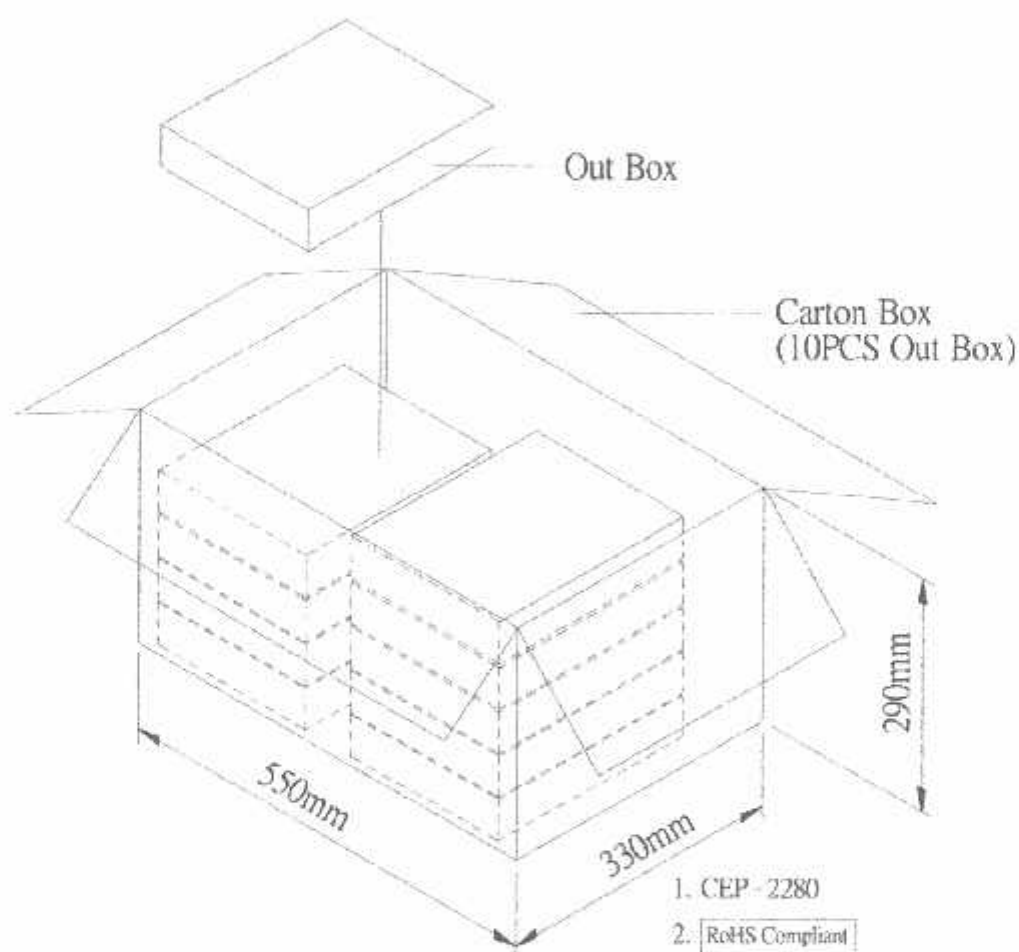
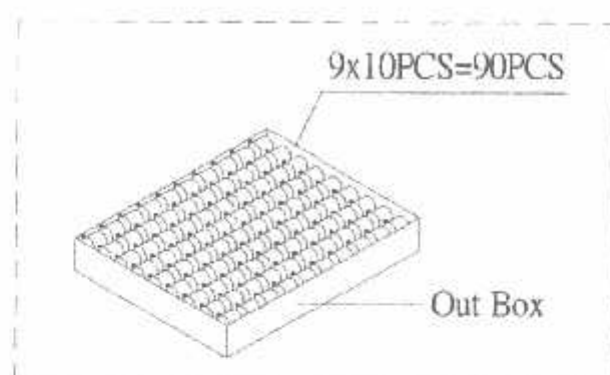
## Life Test

	Test Condition	Evaluation Standard
Life Test	<p>1. Continuous life test: The part will be subjected to 48 hours of continuous operation at +45°C with rated voltage applied.</p> <p>2. Intermittent life test: A duty cycle of 1 minute on, 1 minute off, a minimum of 5,000 times at room temp (+25±2°C) with rated voltage applied.</p>	The buzzer will be measured after being placed at +25°C for 4 hours. The value of the oscillation frequency/current consumption should be ±10% compared to the initial measurements. The SPL should be ±10dB compared to the initial measurements.

## Conditions

Test Condition	a) Temperature: +5 ~ +35°C	b) Humidity: 45 - 85%	c) Pressure: 860 - 1060 mbar
Test Condition	a) Temperature: +25 ±2°C	b) Humidity: 60 - 70%	c) Pressure: 860 - 1060 mbar

aging



Out Box	310mmx248mmx49mm	1x90PCS=90PCS
Carton Box	550mmx330mmx290mm	90PCSx10=900PCS

---

---